ESPECIFICACIÓN DE AUTENTICACIÓN Y FEDERACIÓN DE IDENTIDAD PARA HPC COMO SERVICIO

GISELL GONZÁLEZ ARIAS
ELKIN REINEL MANTILLA GONZÁLEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2014

ESPECIFICACIÓN DE AUTENTICACIÓN Y FEDERACIÓN DE IDENTIDAD PARA HPC COMO SERVICIO

GISELL GONZÁLEZ ARIAS
ELKIN REINEL MANTILLA GONZÁLEZ

Trabajo de Grado para optar al título de Ingenieros de Sistemas

Director
PH.D. CARLOS JAIME BARRIOS HERNÁNDEZ
Co-director
SERGIO AUGUSTO GÉLVEZ CORTÉS

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2014

DEDICATORIA

A Dios por darme la oportunidad cada día de vivir y permitir que esté culminando esta etapa tan importante en mi vida. A mis familiares, quiénes además de ser mis mejores amigos y compañeros; son los pilares que me han dado base firme y sólida para alcanzar esta importante meta. A mi compañera Yesica Gualdrón, por su apoyo constante e incondicional a lo largo de este tiempo, brindándome aliento para continuar con mi labor. A mis amigos (as), por ser más que compañeros de aprendizaje, por ser personas incondicionales y que siempre me brindaron la compañía y el apoyo necesario para alcanzar esta meta. Además expreso mis más sinceros agradecimientos a los docentes, quienes gracias a su colaboración y orientación hicieron realidad la idea que un día fue fruto de nuestra imaginación, y que hoy se convierte en un proyecto de gran valor y crecimiento institucional. A la Universidad Industrial de Santander – UIS y al Centro de Supercomputación y Cálculo Científico – SC3, por abrirme las puertas y permitirme recibir formación profesional, gracias porque además de ser una institución de educación, se ha convertido en un segundo hogar a lo largo de estos años de aprendizaje

Elkin R. Mantilla González

A Dios por darme sabiduría en cada etapa de mi vida
A mi Padre, Madre y Hermanas por ser la principal razón de siempre para seguir adelante y no desfallecer
A mis amigos y compañeros, por los todos los gratos momentos
A esa persona especial, por su compañía y agregarle tranquilidad a mi vida
A los profesores y a la UIS por formarme profesional y personalmente
A mis compañeros de oficina por el apoyo incondicional
Muchísimas gracias a todos.

Gisell González Arias

CONTENIDO

INTRODUCCIÓN ..... - 21 -
2. DESCRIPCIÓN DEL PROYECTO ..... - 23 -
2.1 JUSTIFICACIÓN ..... - 23 -
2.2 DESCRIPCIÓN DEL PROBLEMA ..... - 24 -
2.3 OBJETIVOS ..... - 24 -
2.3.1 Objetivo General ..... - 24 -
2.3.2 Objetivos Específicos ..... - 25 -
3. MARCO TEORICO ..... - 26 -
3.1 HPC (COMPUTACIÓN DE ALTO RENDIMIENTO) ..... - 26 -
3.1.1 Clúster ..... - 27 -
3.1.1.1 Características ..... - 28 -
3.1.1.2 Categorías del Clúster ..... - 29 -
3.1.1.3 Clúster de Alto Rendimiento ..... - 29 -
3.1.2 Cloud Computing o Computación en la Nube ..... - 32 -
3.1.2.1 Capas del Cloud ..... - 33 -
3.1.2.2 Componentes del Cloud ..... - 34 -
3.1.2.3 Modelos de Servicios ..... - 35 -
3.1.2.4 Modelos de Implementación ..... - 37 -
3.1.3 Grid Computing ..... - 38 -
3.1.3.1 Arquitectura del Grid ..... - 40 -
3.1.3.2 Gestor de Recursos y Trabajos ..... - 42 -
3.1.3.3 Grid vs. Cloud ..... - 43 -
3.2 FEDERACION DE IDENTIDAD ..... - 44 -
3.2.1 Componentes de una Federación ..... - 45 -
3.2.2 Proveedor de Identidad ..... - 47 -
3.2.3 Proveedor de Servicio ..... - 48 -
3.2.4 WAYF / DS ..... - 48 -
3.2.5 LDAP ..... - 49 -

3.2.6 Interacción entre Componentes de una Federación ..... - 49 -
3.3 UTILIZANDO SSH SOBRE LA FEDERACION DE IDENTIDAD: FEDSSH - 52 -
3.3.1 Proceso de Autenticación ..... - 53 -
3.3.2 Requisitos ..... - 53 -
3.3.3 Caso de Uso ..... - 54 -
4. ESTADO DEL ARTE ..... - 56 -
4.1 MICROSOFT/IBM ..... - 56 -
4.2 OASIS ..... - 57 -
4.3 LIBERTY ALLIANCE ..... - 59 -
4.4 PAPI ..... - 60 -
5. METODOLOGÍA ..... - 62 -
6. IMPLEMENTACIÓN ..... - 69 -
6.1 ESPECIFICACIÓN Y DEFINICIÓN DEL ESQUEMA ..... - 69 -
6.2 DEFINICIÓN DEL HARDWARE ..... - 72 -
6.3 DEFINICION DEL SOFTWARE ..... - 73 -
7. PRUEBAS Y ANALISIS DE RESULTADOS ..... - 80 -
7.1 PRUEBA LOCALMENTE Y DE LA APLICACIÓN FEDSSH ..... - 80 -
7.2 PRUEBA EXTERIOR TESTSHIB ..... - 81 -
8. CONCLUSIONES ..... - 83 -
9. RECOMENDACIONES ..... - 85 -
10. LIMITACIONES ..... - 86 -
REFERENCIAS BIBLIOGRAFÍCAS ..... - 87 -
BIBLIOGRAFÍA ..... - 89 -
ANEXOS ..... - 91 -

LISTA DE FIGURAS

FIGURA 1: CONFIGURACIÓN DE UN TÍPICO CLÚSTER 29

FIGURA 2: COMPUTACIÓN LOCAL A UN NODO DE CLÚSTER 31

FIGURA 3: COMPUTACIÓN PARALELA 32

FIGURA 4: CAPAS DEL CLOUD 34

FIGURA 5: COMPONENTES DEL CLOUD 35

FIGURA 6: ESQUEMA DE SAAS 36

FIGURA 7: GRID COMPUTING 39

FIGURA 8: ARQUITECTURA DE CAPAS DEL GRID 41

FIGURA 9: ESQUEMA DEL FUNDAMENTO BÁSICO DE FEDERACIÓN DE IDENTIDADES 45

FIGURA 10: COMPONENTES DE UNA FEDERACIÓN 47

FIGURA 11: INTERACCIÓN ENTRE COMPONENTES (PASOS DEL 1 AL 3) 50

FIGURA 12: INTERACCIÓN ENTRE COMPONENTES (PASOS DEL 4 AL 8) 52

FIGURA 13: CASO DE USO 55

FIGURA 14: PROCESO METODOLÓGICO 62

FIGURA 15: REPRESENTACIÓN DEL PRIMER ESQUEMA 70

FIGURA 16: ESQUEMA FINAL 71

FIGURA 17: PROVEEDOR DE IDENTIDAD DEL PROTOTIPO IMPLEMENTADO 74

FIGURA 18: PROVEEDOR DE IDENTIDAD DE SHIBBOLETH 75

FIGURA 19: PROVEEDOR DE SERVICIOS DEL PROTOTIPO IMPLEMENTADO 76

FIGURA 20: FUNCIONAMIENTO DE SIMPLESAML 77

FIGURA 21: RESULTADO PRUEBA LOCALMENTE Y APLICACIÓN FEDSSH 80

FIGURA 22: RESULTADO PRUEBA EXTERIOR TESTSHIB 81

LISTA DE TABLAS

TABLA 1: ESPECIFICACIONES DE HARDWARE 73

TABLA 2: ESQUEMAS 78

TABLA 3: DIRECCIONES IP DEL PROTOTIPO 79

LISTA DE ANEXOS

ANEXO A: CONFIGURACIÓN SERVIDOR IDP SHIBBOLETH 89

ANEXO B: CONFIGURACIÓN SERVIDOR SP SIMPLESAML 100

ANEXO C: CONFIGURACIÓN DE LA APLICACIÓN FEDSSH 103

ANEXO D: PARCHE Y CONFIGURACIÓN OPENSSH 108

ACRÓNIMOS

HPC	High-Performance Computing
SP	Service Provider
IDP	Identity Provider
WAYF	Where Are You From
LDAP	Lightweight Directory Access Protocol
SSO	Single Sign-On
HPCaaS	High-Performance Computing as a Service
AaaS	Application as a Service
SaaS	Software as a Service
PaaS	Platform as a Service
IaaS	Infrastructure as a Service
LAN	Local Area Network
NFS	Network File System
NIST	National Institute of Standards and Technology
CPU	Central Processing Unit
API	Application Programming Interface
SDK	Software Development Kit

SSH	Secure Shell
IBM	International Business Machines
PAPI	Point of Access to Providers of Information
OASIS	Organization for the Advancement of Structured Information Standandards
SAML	Security Assertion Markup
URL	Uniform Resource Locator
HTTP	Hypertext Transfer Protocol
XML	Extensible Markup Language

GLOSARIO

Supercomputación: Campo de la ingeniería de sistemas orientada a optimizar la duración en la ejecución de tareas, lo que permite que el tiempo de cálculo disminuya significativamente.

Clúster: Arreglo de computadores que utilizan componentes comunes y están conectados por una red para trabajar en conjunto en cierto gran problema para que este pueda ser resuelto en pequeños pedazos.

Cloud (nube): Es un modelo de servicio de cómputo, fundamentalmente en tres niveles (usuario, plataforma e infraestructura), conformado por servidores virtuales que ofrecen recursos computacionales disponibles en todo internet, a los cuales se puede acceder a través de aplicaciones web.

Grid: El Grid Computing es una tecnología de computación distribuida que integra y coordina recursos informáticos principalmente heterogéneos, como el cómputo, el almacenamiento y las aplicaciones.

Federación de Identidad: Conjunto de organizaciones que forman un círculo de confianza que facilita la autenticación y autorización entre diferentes dominios, en donde ofrecen sus servicios entre ellas para mejorar los beneficios de sus usuarios de manera segura.

Atributos: Son un conjunto de datos, características referentes a un determinado usuario que el servicio requiere para utilizar el acceso a los recursos.

Recursos: Datos custodiados por el proveedor de servicios a los cuales se podrá acceder mediante la respectiva autenticación y autorización.

Single Sing-On: Se basa en que el usuario se auténtica una única vez en una de las organizaciones y a partir de ahí, puede acceder al resto de sistemas sin tener que volver a autenticarse.

Alto Rendimiento: Efectividad del desempeño de una computadora sobre una aplicación.

Autorización: Es el control de acceso, mecanismo en el cual se permite el acceso a los bienes que se están protegiendo.

Autenticación: Proceso por el cual un sistema, a través de un mecanismo determinado constata que el usuario es quien dice ser.

Usuario: Es cualquier elemento capaz de presentarse a través de medios electrónicos, al cual se le presentarán servicios después de realizar la respectiva autenticación.

Nodo: Punto de conexión de una red, normalmente un computador, el cual tiene recursos específicos y una especial importancia para más de un usuario.

Ancho de Banda: Indica la máxima cantidad de información simultánea que puede transferir, entre mayor sea mejor.

Programación paralela: Forma de cómputo en la que muchas instrucciones se ejecutan simultáneamente, operando sobre el principio de que problemas grandes, a menudo se pueden dividir en problemas más pequeños, que luego son resueltos simultáneamente sin depender uno de otro.

Servidor: Sistema informático que contiene datos o archivos, que son

proporcionados como servicios a usuarios remotos (clientes) u otros servidores.

Redes: Sistema de elementos interrelacionados que se conectan para proporcionar una comunicación local o remota y permitir el intercambio de recursos e información entre usuarios con interés común.

Navegador: Aplicación para acceder y visualizar documentos www. y navegar en internet

Protocolo: Descripción formal de formatos de mensajes y de reglas que se deben seguir para intercambiar correctamente esos mensajes.

Web Service: Servicio Disponible a través de HTTP/HTTPS

Aplicación: Forma de llamar a los programas que nos permiten trabajar con el computador

RESUMEN

Título:
ESPECIFICACIÓN DE AUTENTICACIÓN Y FEDERACIÓN DE IDENTIDAD PARA HPC COMO SERVICIO[1]

Autores: GONZÁLEZ ARIAS, Lidys Gisell, MANTILLA GONZÁLEZ, Elkin Reinel[2]

Palabras Claves: Federación de identidades, Single Sing On, Secure SHell, Login, HCPaaS.

DESCRIPCIÓN

Cada vez es más importante para las organizaciones e instituciones formar convenios para aumentar los beneficios que son ofrecidos a sus usuarios, para que estos convenios puedan funcionar es necesario que cada entidad se encargue de las autenticaciones de los usuarios que la conforma y que se cree un ambiente de confianza entre las entidades. Esto se hace para evitar autenticaciones extras cada vez que los usuarios pertenecientes a las entidades del convenio, deseen ingresar a los servicios ofrecidos. Esto se conoce como Single Sing On (SSO), el cual ofrece una sola autenticación en un solo punto.

La federación de identidades es el esquema que se encarga de que los convenios entre las entidades funcionen, ya que en una federación de identidades cada organización es responsable de la gestión de usuarios y sus atributos, como del acceso a los recursos. Por esta razón cada usuario tiene acceso a todos los recursos ofrecidos por las entidades que conforman la federación, mediante una sola autenticación.

Los servidores SSH (Secure SHell) con infraestructura federada realizan la autenticación, basándose en mecanismos de llaves públicas de sus usuarios. En donde los usuarios solo deben recordar una sola contraseña y generar una llave pública almacenada en sus dispositivos para que de esta forma tengan acceso a HPCaaS mediante SSH con un solo login.

La idea principal de este proyecto es llevar a cabo la presentación de un esquema de federación de identidad para compartir los recursos de HPCaaS de la Universidad Industrial de Santander (UIS), administrados por el Centro de Supercomputación y Cálculo Científico o permitiendo SSO con Clave pública en protocolo SSH.

[1] Trabajo de Grado con modalidad de Trabajo de Investigación.
[2] Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Carlos Jaime Barrios Hernández, Ph,D

ABSTRACT

Title:
A SPECIFICATION AND IDENTITY FEDERATION FOR HPC AS A SERVICE [3]

Authors: GONZÁLEZ ARIAS, Lidys Gisell, MANTILLA GONZÁLEZ, Elkin Reine[4]l

Keywords: Federation of identities, Single Sing On, Secure SHell, Login, HCPaaS.

DESCRIPTION

Nowadays, the federation of identity is important for the organizations and institutions to form agreements to increase the benefits that are offered to their users. These agreements work during the authentication process, when the user access to the entities of the agreement to have offered services without extra-authentication. This process is known like Sing Sail On (SSO), which offers a single authentication in a single point.

The federation of identities is the scheme that takes charge that the agreements among the entities function. Then, the federation is the responsible of the administration of users and their attributes, as is the case of the resources access. This point is important, because is for this reason that each user has only one access to all resources offered by the entities. In other words, the entities inside the federation are accessed by a single authentication.

SSH (Secure SHell) servers with federated infrastructure they carry out the authentication based on mechanisms of users public keys. In the case of High Performance as a Service (HPCaaS), the single login reduce the complexity of the resources access to a single password associated to a public key stored in the devices of the resources.

This project aims to carry out an scheme of identity federation to share resources in a Paces model managed by the High Performance and Scientific Computing Centre of the Industrial University of Santander (UIS) allowing SSO with public Key in protocol SSH.

[3] Bachelor Thesis Research Paper mode..
[4] Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Carlos Jaime Barrios Hernández, Ph.D

INTRODUCCIÓN

Desde hace unos años el deseo de encontrar soluciones a problemas con un alto nivel de complejidad en menor tiempo y disponer de alta capacidad computacional, se ha convertido en el objetivo de ingenieros y científicos, esta meta ha llevado al desarrollo de componentes de hardware y software que han hecho posible la creación de tecnologías como clúster, súper-computadoras y programación en paralelo, hasta llegar a la utilización de tecnologías de cloud, así como el servicio HPC (Infraestructura como servicio) el cual nos proporciona los recursos de computación necesarios.

La Universidad Industrial de Santander con su infraestructura “GridUIS-2” que integra diferentes recursos tecnológicos entre los cuales se encuentra Clúster y una Supercomputadora (Guane-1) distribuido dentro del campus central y la sede Guatiguará dando así servicio de HPC a la comunidad universitaria y científica. Sin embargo con el deseo de la universidad a expandir sus servicios para poder tener acceso a más capacidad de cómputo y que otros usuarios puedan tener acceso a los recursos de la institución, gracias a convenios con otras universidades dentro y fuera del país, por ejemplo organización Grid5000; dando así la posibilidad de acceso a recursos de cómputo.

De la necesidad de compartir estos recursos informáticos entre distintas organizaciones y el hecho de hacer login en cada una de las instituciones para tener acceso a los recursos, implicaba para los usuarios algo tedioso. Actualmente se está optando por acuerdos entre las organizaciones en la cual estas comparten la identidad de los usuarios con sus (respectivos) atributos. A partir de lo anterior se introduce el concepto Federación de Identidad en la cual el

usuario tendrá una sola identificación y es así posible compartir tanto la información del usuario como los servicios entre los componentes de la federación. Se dice que un usuario posee una “identidad federada” cuando este puede navegar o utilizar recursos de otras organizaciones que componen la federación necesitando loguearse solo en una de ellas, esto se conoce como SSO (Single Sing-On por sus siglas en inglés).

En este trabajo se presenta una propuesta de un esquema de federación viable para compartir recursos HPCaaS permitiendo SSO con Clave pública en protocolo SSH , en la primera parte presentaremos la justificación y los objetivos de la propuesta, seguida de los conceptos y los proyectos afines existentes; en tercer lugar presentamos la metodología de trabajo, implementación, pruebas y conclusiones .

2. DESCRIPCIÓN DEL PROYECTO

2.1 JUSTIFICACIÓN

Dentro de la arquitectura de visibilidad de los usuarios ante los servicios de computación en la nube, se identifican diferentes niveles que involucran utilización de aplicaciones y software (AaaS, SaaS), Plataformas (PaaS) e Infraestructura (IaaS). Hoy en día, en diferentes comunidades tanto empresariales, como de gobierno o incluso académicas, requieren utilizar ambientes que garanticen recursos de altas prestaciones. Esa oferta de servicios se conoce comúnmente como HPC como Servicio (del inglés HPC as a Service: HPCaaS), el cual involucra el acceso a recursos tanto de infraestructura como de plataforma.

Para garantizar el uso autorizado de esos recursos garantizando no solo la confidencialidad sino la preservación de los procesos y de la información, es necesaria una gestión de identidad y un manejo de protocolos y metodologías de seguridad, que correspondan a un esquema específico.

Teniendo en cuenta diferentes plataformas de cómputo de alto rendimiento y diferentes comunidades/organizaciones que interactúan con estas plataformas en cualquiera de los niveles de visibilidad de computación en la nube, además de la interoperabilidad con otros proveedores de servicios, el problema de la gestión de identidad y federación asociada, es un problema abierto, que involucra tanto aspectos técnicos como metodológicos importantes, que se tratarán en este proyecto.

2.2 DESCRIPCIÓN DEL PROBLEMA

Desde hace algún tiempo las organizaciones, para brindar mayores beneficios a los usuarios, tomaron la decisión de aumentar la seguridad, enfocándose principalmente en el servicio de la autenticación y de los atributos únicos de cada usuario. A raíz de estas mejoras, la idea de realizar un convenio para compartir recursos con otras organizaciones no era un camino muy viable, debido a que se presentaron problemas a falta de un estándar de identificación que estimulará la puesta en común de recursos, información y servicios para las organizaciones.

Al querer un usuario ingresar a los recursos a los que tiene autorización, debe autenticarse con una credencial asignada previamente por la organización a la cual pertenece. A su vez, si este usuario necesita acceder a los recursos de cualquier otra organización con la que exista convenio, éste debe identificarse en cada una de estas, con una credencial de identificación distinta asignada por el servidor de cada organización. Tener una credencial por usuario en todos los servidores, implica para cada organización un gasto de infraestructura y económico.

2.3 OBJETIVOS

2.3.1 Objetivo General

Concebir y proponer un esquema de seguridad que provea HPC como Servicio, teniendo en cuenta lineamientos de autenticación y federación de identidad.

2.3.2 Objetivos Específicos

1. Analizar y definir un esquema de seguridad que permita el acceso a los recursos de cómputo de la institución, tanto propio como compartido, teniendo en cuenta aspectos como la autenticación y federación de identidad.

2. Definir la estrategia tecnológica para la interacción segura con los recursos observan tres aspectos: usuarios-plataformas, usuarios-aplicaciones y aplicaciones-plataformas.

3. Evaluar la estrategia tecnológica seleccionada mediante un prototipo que permita observar dos casos posibles: interacción a través de servicios privados (la misma organización) e interacción a través de servicios públicos (dos organizaciones diferentes)

3. MARCO TEORICO

3.1 HPC (COMPUTACIÓN DE ALTO RENDIMIENTO)

HPC (High Performance Computing) o Computación de Alto Rendimiento, es la solución tecnológica desarrollada para soportar la demanda de aplicaciones computacionales que requieren un alto nivel de procesamiento para obtener resultados en el menor tiempo posible. Usualmente las aplicaciones que utilizan tecnología HPC para obtener resultados, son aplicaciones de desarrollo científico elaboradas por grupos de investigación u organizaciones enfocadas a diferentes ramas científicas.

Desde el punto de vista del Hardware, HPC está conformado por tres tipos de infraestructuras [1]:

- **Clúster:** Grupo de cientos o miles de servidores con alta capacidad de procesamiento y de almacenamiento de datos, ubicados en un mismo lugar, los cuales trabajan juntos para dar solución a un mismo problema. Antiguamente, los supercomputadores eran máquinas de un solo procesador especial, ahora estos son a su vez catalogados como clústers.

- **Cloud Computing:** Una nube HPC proporciona el acceso a los recursos de forma escalable y dinámica, empleando la web como medio de comunicación y utiliza como base el modelo de computación como servicio (a-as-service).

- **Grid Computing:** Grupo de servidores distribuidos que trabajan en conjunto para ofrecer mayores recursos a las organizaciones que la conforman.

3.1.1 Clúster

Se habla de clúster, cuando un grupo de computadores con hardware homogéneo (lo cual no es obligatorio) y software especializado a nivel de aplicación o núcleo están conectados a una red local (LAN); Como característica principal, el clúster se comporta como si fuese un único computador con alto nivel de procesamiento, que trabaja en forma paralela para obtener resultados en poco tiempo.

Un clúster puede generar tres tipos de tráficos [2]: Tráfico de Computación, el cual se da entre nodos de computación; Tráfico del sistema de Archivos, usualmente un servidor NFS (Network File System); y Monitoreo de Tráfico, que proporciona un nodo de monitoreo y control de trabajo en todo el clúster, no es obligatorio que exista un nodo de monitoreo en un clúster, ya que éste puede ser reemplazado por un frontend o puede haber algún tipo de monitoreo para todo un sitio multi-clúster.

Un clúster puede estar diseñado con: múltiples nodos de servidores de archivos, múltiples usuarios de nodos de acceso y nodos de administración independiente.

Un núcleo es la unidad de procesamiento dentro de un procesador moderno, mas núcleos por nodo significa más procesamiento por nodo, ya que cada núcleo podría utilizarse para ejecutar un programa totalmente independiente; Y los hilos, son las rutas de ejecución concurrentes.

3.1.1.1 Características

- **Nodo Principal o Maestro**

Es la puerta de entrada a una red compartida, por la cual se puede conectar al clúster. El nodo principal, tiene una o más redes por medio de las cuales se comunica a los nodos de trabajo, estas redes son privadas y solo se puede acceder a ellas a través del clúster.

Contiene gran cantidad de almacenamiento que es compartida a través de la red por todos los nodos de trabajo.

- **Nodos de Trabajo**

Son los nodos encargados de realizar el trabajo de la computación. Estos nodos son homogéneos, es decir, que son casi siempre idénticos en hardware y software, y se comunican entre sí y con el nodo principal en las redes privadas.
La cantidad de nodos de trabajo en un clúster es variable pero puede ser una cantidad muy grande.

Para mantener ocupados todos los nodos, se necesita una buena conexión entre los dispositivos, la cual se clasifica según su Latencia, que corresponde al menor tiempo en el que un byte se puede enviar, y Ancho de Banda (Máxima velocidad de datos).

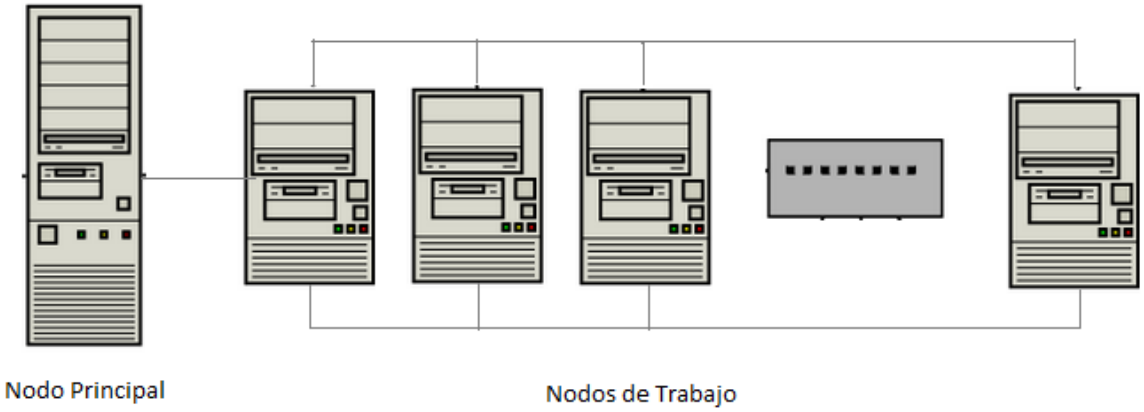

Figura 1. Configuración de un Típico Clúster

Fuente : Autores

3.1.1.2 Categorías del Clúster

- **Capabilidad (Capability)**

Un clúster de capabilidad [3] está diseñado para manejar trabajos grandes de cómputo que empleen todos los nodos del clúster.

- **Capacidad (Capacity)**

Este tipo de clúster [3] se utiliza para suministrar una cierta cantidad de capacidad de cómputo a los usuarios finales. Los clúster con esta categoría, pueden soportar cientos de usuarios que utilizan cierto número de trabajos, los cuales necesitan un número menor de nodos.

3.1.1.3 Clúster de Alto Rendimiento

Un clúster de alto rendimiento, funciona óptimamente para problemas grandes y

complejos que requieren de gran cantidad de potencia computacional y para los cuales el tiempo de solución constituye un papel importante. Está formado por un número de máquinas individuales que trabajan como si fuesen una sola maquina muy potente.

El clúster puede trabajar en forma de “Telaraña”, ya que un trabajo determinado no se ejecuta por un solo nodo de trabajo, en lugar de esto, es enviado a todo el clúster para realizar su ejecución.

El Clúster administra los recursos necesarios para ejecutar un determinado trabajo y asigna los trabajos a un puesto en la cola de trabajos. Dependiendo de los recursos, todos los trabajos independientes, es decir los trabajos en los que no existe comunicación entre la ejecución de cada uno de ellos, pueden ejecutarse simultáneamente, aunque algunos esperan en la cola mientras que otros finalizan su ejecución. Este tipo de computación es *Computación Local a un nodo de clúster* [4], lo que significa que el nodo no se comunica con otros nodos, pero puede necesitar el acceso al sistema de archivos.

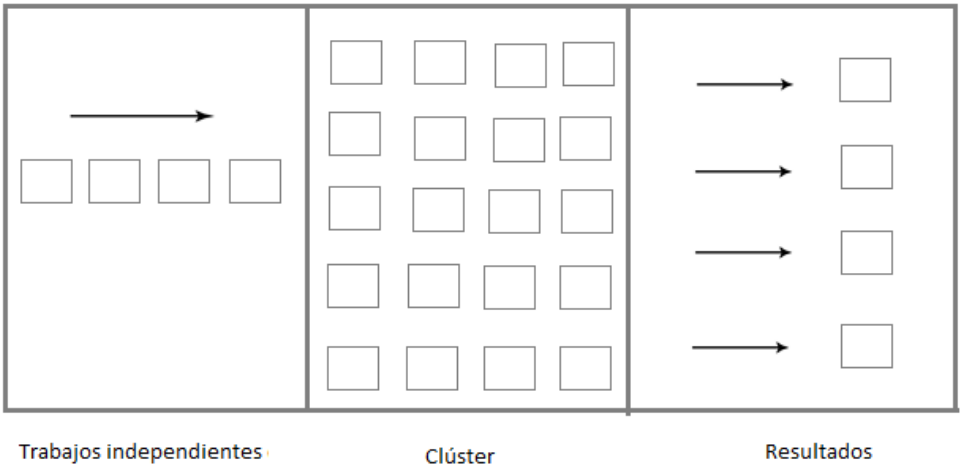

Figura 2. Computación Local a un Nodo de Clúster
(Se ejecutan de manera simultánea trabajos independientes)
Fuente: Autores.

Otra forma de ejecución del clúster de alto rendimiento, consiste en que el clúster divide un trabajo HPC en varios sub-trabajos de un tamaño menor y ejecuta estos pequeños sub-trabajos en diferentes nodos. Este proceso de división se realiza a nivel se software.

Debido a que los sub-trabajos necesitan comunicarse entre ellos, se puede causar tráfico de cómputo en el clúster.

En este tipo de ejecución, es común utilizar miles de nodos para obtener el resultado de un solo trabajo, esto se conoce como *Computación Paralela,* lo que significa que un programa paralelo (es decir, un programa en el que se puede utilizar un software para dividirlo) solicita al clúster los recursos necesarios para ejecutarse, una vez realizada la división del programa y los recursos estén libres, el clúster ejecutará el programa utilizando varios nodos para la solución del mismo.

Existen dos tipos de Computación Paralela: Paralelismo de datos, el cual consiste en la división de un programa en subdominios; donde cada nodo trabaja sobre datos independientes, y Paralelismo de Tareas, en este tipo de computación se identifican las regiones o tareas del programa y solo se ejecutan en paralelo aquellas que son independientes.

Los tipos de memoria de la computación en Paralelo, son: Memoria compartida, en donde los trabajos acceden a la memoria global, y Memoria Distribuida, la cual utiliza comúnmente paso de mensajes para enviar y recibir datos y sincronizar procesos.

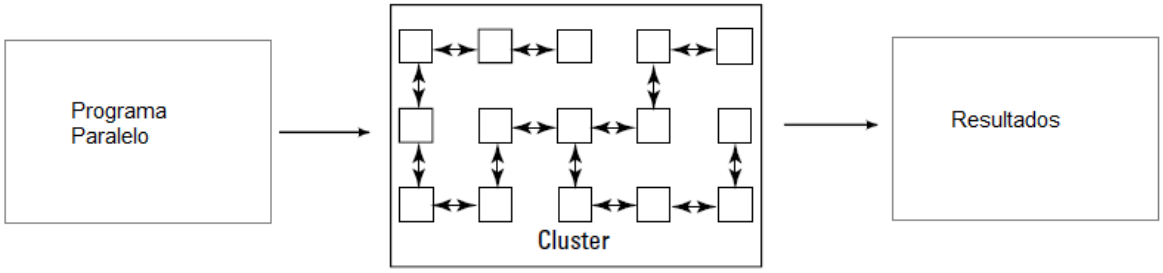

Figura 3. Computación Paralela

Fuente: Autores.

3.1.2 Cloud Computing o Computación en la Nube

Según la definición del NIST [5] (National Institute of Standards and Thecnology, Instituto Nacional de Estándares y Tecnología), el término Cloud se puede interpretar como “un modelo para permitir un acceso ubicuo, conveniente, bajo demanda y a través de la red, a un conjunto compartido de recursos informáticos configurables (como redes, servidores, almacenamiento, aplicaciones y servicios)

que pueden ser rápidamente aprovisionados y liberados con el mínimo esfuerzo de gestión o de interacción con el proveedor del servicio".

Una explicación más sencilla de lo que es el cloud, que es la posibilidad de ofrecer servicios de computación y recursos informáticos sobre internet, siendo transparente a los usuarios. Estos servicios ofrecidos son capaces de ajustarse a las peticiones de los usuarios para su asignación y el consumo de recursos de computo, es decir si un usuario necesita mayor capacidad de almacenamiento, el cloud creará nuevas máquinas virtuales para satisfacer las necesidades del usuario y se eliminan las máquinas virtuales cuando al usuario le sobran recursos, estas asignaciones o ajustes se realizan de forma dinámica.

Para acceder a las aplicaciones del cloud son necesarios exploradores de internet, mientras que el software y los datos son guardados en los servidores, por ejemplo: Un usuario tiene varias cuentas de correo electrónico (Gmail, Hotmail, Yahoo, etc.) para acceder a esas cuentas de correo, no es necesario que se instale ningún tipo de software distinto al del navegador en el computador personal, solo se necesita acceder al navegador, autenticarse en el servidor y así se puede manipular toda la información de la cuenta de correo, la cual no se almacena en el computador, sino que se almacena en los servidores de la empresa que presta el servicio.

3.1.2.1 Capas del Cloud

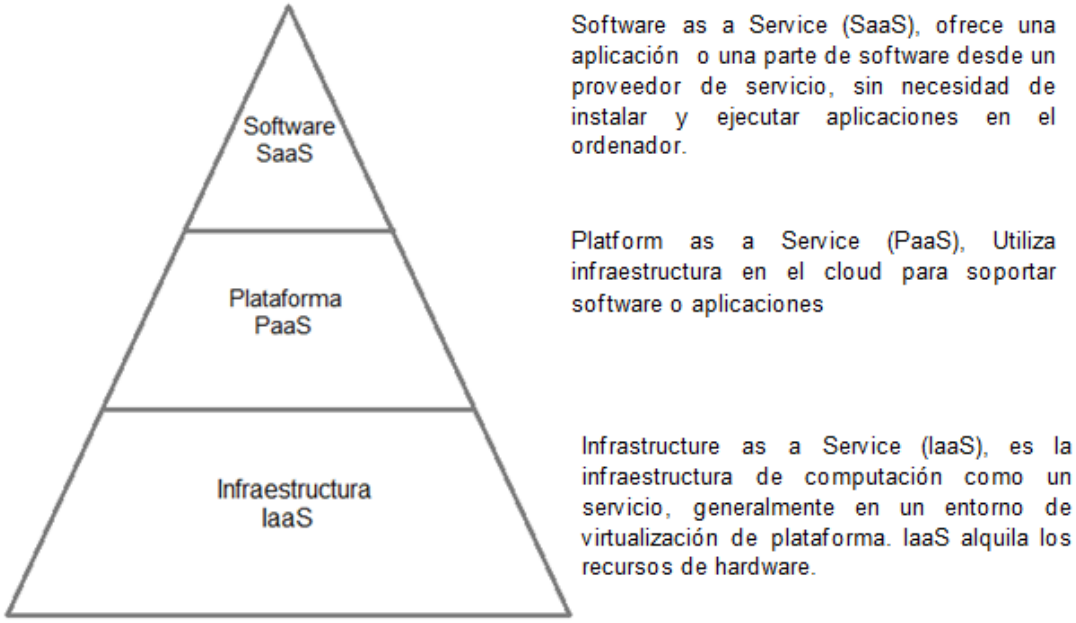

Figura 4. Capas del Cloud

Fuente: Autores.

3.1.2.2 Componentes del Cloud

- **Clientes**

Se le dicen clientes a todos aquellos dispositivos (PC's, Portátiles, Tablet o Teléfonos Móviles) que son manipulados por los usuarios finales para acceder a la información y procesos de la nube.

Existen tres tipos de clientes: Dispositivos Móviles; Clientes Ligeros, son los clientes que no tienen disco duro interno y por esta razón dejan que el servidor haga todo el trabajo, pero al final son capaz de mostrar la información de los resultados; y Clientes Gruesos, son los que utilizan algún tipo de navegador para acceder a la nube.

- **Centro de Datos**

Son los servidores en donde están las aplicaciones a las que se acceden.

- **Servidores Distribuidos**

En estos servidores también están las aplicaciones e información, es decir los centros de datos o partes de centros de datos, a los que se quiere acceder, la diferencia está en que estos servidores no se encuentran ubicados en un mismo lugar, se encuentran dispersos geográficamente.

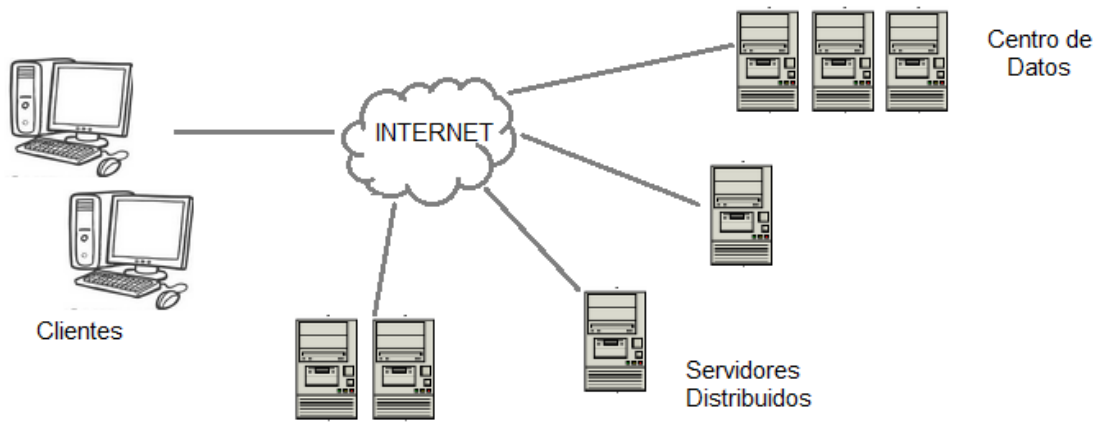

Figura 5. Componentes del Cloud

Fuente: Autores.

3.1.2.3 Modelos de Servicios

- **Software as a Service (SaaS)**

Es la forma en la que una aplicación se presenta como servicio a los clientes que acceden a internet. Como no es necesaria una instalación, el cliente no tiene que realizarle mantenimiento, soporte, ni realizar integraciones con otros sistemas.

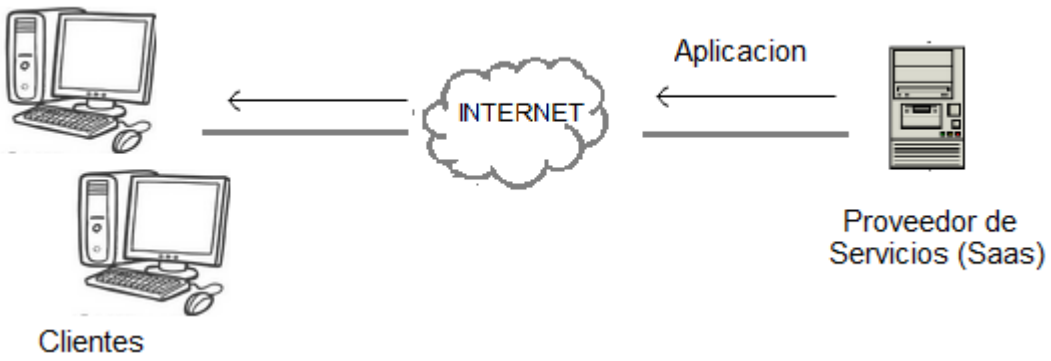

Figura 6. Esquema de SaaS

Fuente: Autores

En otras palabras, SaaS es utilizar un software remoto o herramientas como un servicio ofrecido por medio de un navegador de internet, en lugar de un software instalado localmente.

Los servicios SaaS se gestionan en un lugar central, de esta forma los clientes pueden acceder desde donde tengan acceso a la red. El software que utilizan los clientes es mantenido y soportado por la empresa creadora del software, así mismo la empresa mantendrá la información del usuario en sus bases y proveerá los recursos necesarios.

- **Platform as a Service (PaaS)**

Es un modelo que proporciona los recursos necesarios para construir y ejecutar aplicaciones y servicios web desde internet, sin necesidad de descargar e instalar algún software

- **Infrastructure as a Service (IaaS)**

No ofrece aplicaciones a los clientes, en cambio ofrece el hardware y provisiones de computo (recursos físicos) necesarios para soportar software y aplicaciones

Entre los servicios que IaaS alquila es encuentran: Espacio de Servidor, Equipo de Red, Procesamiento de Datos, Ciclos de CPU, Memoria, Almacenamiento, entre otros.

3.1.2.4 Modelos de Implementación

- **Nube Pública**

En este modelo de nube, la infraestructura y los recursos que hacen parte del servicio se encuentran disponibles para los usuarios por medio de internet y son propiedad y operados por un proveedor de la nube.

- **Nube Privada**

Su infraestructura se gestiona por una sola organización, las aplicaciones y servicios pueden ser gestionados por esta misma organización o por un proveedor de servicios, Es decir; en la nube privada están aquellos servicios o recursos que son exclusivos para una organización.

- **Nube Comunitaria**

Su infraestructura y servicios son compartidos y exclusivos para varias organizaciones que tengan objetivos similares. Es decir, solo funciona para este grupo de organizaciones.

- **Nube Híbrida**

Es la combinación de dos o más tipos de implementación de nubes, que se mantienen como entidades separadas pero están unidas por tecnologías estandarizadas que hacen posible la portabilidad de datos y aplicaciones.

3.1.3 Grid Computing

El Grid [6] es un sistema de computación distribuida (Infraestructura), a través de la cual se forma una supercomputadora con gran potencia de cálculo y almacenamiento, compuesta por un grupo de computadoras de alto rendimiento, redes y bases de datos, que se encuentran conectados a la red libremente, a los cuales se acceden remotamente y trabajan en conjunto para realizar tareas complejas compartiendo potencia computacional y una gran cantidad de recursos.

En resumen, el Grid Computing es un conjunto de computadoras de diferentes instituciones conectadas en una red que utilizan sus recursos para trabajar en un solo problema al mismo tiempo.

Todos los recursos de un Grid necesitan una interconexión de hardware especial y un software que controle estos recursos para ensamblarlos en el Grid, ya que un Grid no está conformado por una sola organización utilizando sus recursos, sino que intervienen varias organizaciones geográficamente separadas cada una utilizando, compartiendo y virtualizando sus recursos computacionales. De esta forma, si una organización en un momento dado tiene todos sus recursos trabajando, y necesita aún más de los que dispone, es función del Grid equilibrar la utilización de sus recursos, buscando maquinas remotas, las cuales no estén trabajando y puedan ejecutar los trabajos de la organización que los necesita.

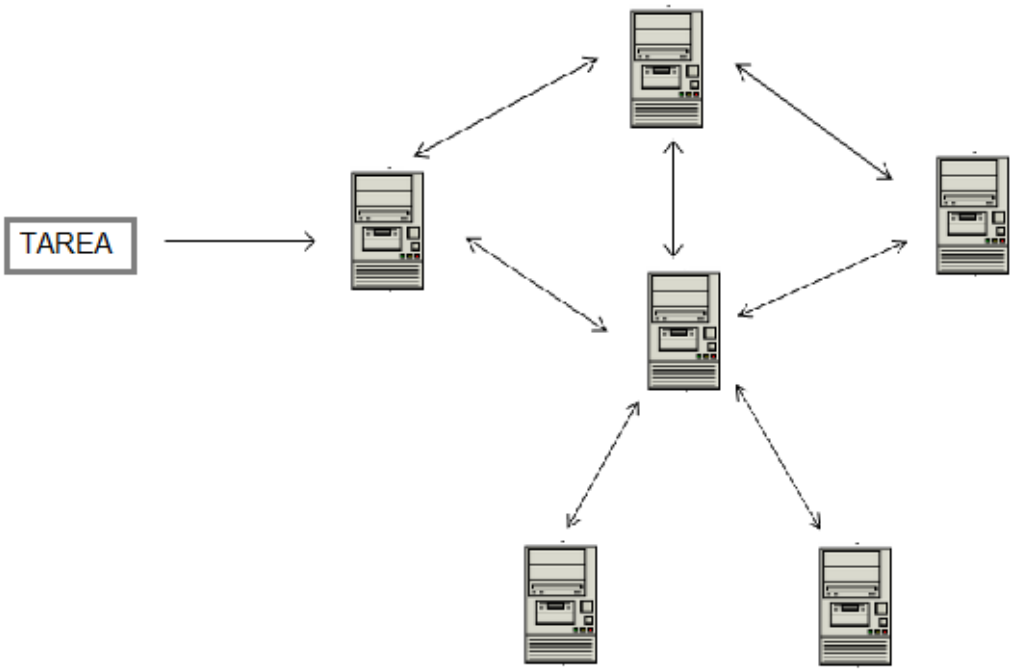

Figura 7. Grid Computing

Fuente: Autores

El Grid computing utiliza un software para dividir y enviar partes independientes (ayudando a la escalabilidad) de un mismo programa a miles de computadoras

para su ejecución. Además de compartir recursos para trabajar en un solo problema, una infraestructura Grid tiene: Fiabilidad y Alta Disponibilidad de sus recursos, de esta forma, se reducen los tiempos de ocio aprovechando los ciclos de procesamiento y proporcionando un mejor soporte a las aplicaciones; también organiza a los usuarios en organizaciones virtuales para facilitar el proceso de autorización cuando los usuarios acceden a el Grid.

3.1.3.1 Arquitectura del Grid

Es una arquitectura abierta de definición de protocolos y servicios, los cuales son los encargados de establecer y controlar las relaciones entre las organizaciones que forman el Grid y sus recursos computacionales. Es una arquitectura abierta constituida por: Interfaces de Programación de Aplicaciones (API's) y Herramientas para el Desarrollo de Software (SDK's).

Esta arquitectura está organizada en cinco capas [7]: Infraestructura, Conectividad, Recurso, Recursos y Aplicación

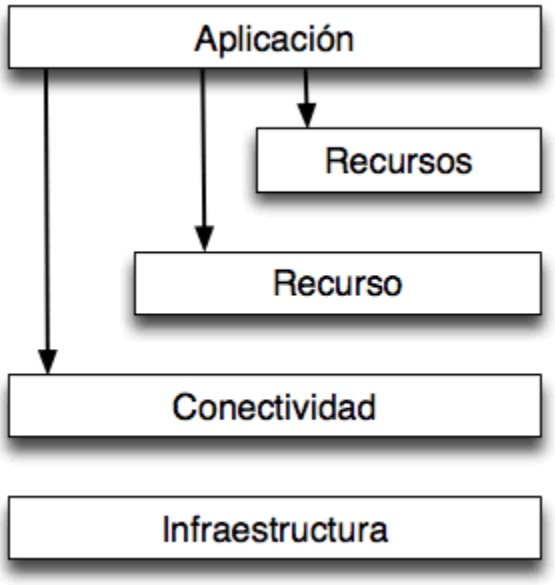

Figura 8. Arquitectura de capas del Grid

Fuente: Autores

- **Infraestructura**

Se encuentran los recursos computacionales que serán compartidos por las organizaciones

- **Conectividad [8]**

Se encuentran protocolos estándar de seguridad y comunicación que permite a los recursos comunicarse. Los protocolos de comunicación permiten el intercambio de datos entre la capa de Infraestructura y los recursos; los protocolos de seguridad brindan mecanismos de criptografía para identificar usuarios y recursos.

- **Recurso**

Se encuentran los protocolos que se enfocan en gestionar un único recurso y permiten tener información y control del mismo.

- **Recursos**

Abarca todos los protocolos y servicios que gestionan la interacción de un conjunto de recursos.

- **Aplicación**

Se centra en la definición de protocolos que controlan el acceso de las aplicaciones a la Infraestructura Grid a través de las capas. Son las aplicaciones de los usuarios que operan en el Grid.

3.1.3.2 Gestor de Recursos y Trabajos

Es uno de los servicios ofrecidos por la capa Recursos, este servicio reacciona según la disponibilidad actual de los recursos en el Grid, evitando los tiempos de ocio y es el encargado de asignar las tareas a cada recurso encontrando la maquina apropiada para ejecutarla.

Existen tres componentes de Scheduler: Planificador de Trabajos, maximiza la cantidad de trabajo; Planificador de Recursos, maximiza el uso de recursos; Planificador de Aplicación, divide la aplicación en tareas, asigna recursos a las tareas y se asegura de su desarrollo.

3.1.3.3 Grid vs. Cloud

- **CLOUD = GRID + WEB SERVICE**

- **Arquitectura:**

Básicamente ambos modelos proponen el agrupamiento de los recursos conectados a Internet, con el fin de proveer una vista abstracta de una única facilidad capaz de proveer diversos recursos computacionales. Es por eso que en ambas arquitecturas son necesarias al menos tres capas: una capa inferior encargada de gestionar los recursos de manera directa, una capa media que actúe como intermediario entre los recursos y las aplicaciones, y por último una capa superior que provea las aplicaciones que los usuarios necesitan para poder utilizar el sistema.

- **Accesos:**

En la Grid, para tener acceso a los recursos, se inicia sesión en el shell de controlador del mismo (ssh root@controllerhost), Una vez iniciada la sesión correctamente, se encontrará en el shell Bash de Linux.
Por el contrario, en la Cloud todo lo que puede ofrecer un sistema informático se ofrece como servicio, de modo que los usuarios pueden acceder a los servicios disponibles "en la nube de Internet" mediante una conexión a internet desde cualquier dispositivo móvil o fijo ubicado en cualquier lugar, sin conocimientos (o, al menos sin ser expertos) en la gestión de los recursos que usan [9].

- **Modelo de seguridad:**

Ambos sistemas ofrecen su capacidad de almacenamiento y cómputo a otros sitios o usuarios que lo requieran.

Con respecto a Cloud, Aquellas organizaciones virtuales que implementen el Cloud serán las responsables de proveer al menos un contrato de confidencialidad en el contexto de procesamiento o almacenamiento de información.

En contraste, en Grid, dado que el objetivo es la cooperación en lo que a gestión de recursos respecta, la utilización de protocolos y estándares resulta ser el enfoque utilizado, por lo que este tipo de sistemas posee un mayor grado de confiabilidad. En este sentido, en Grid existe una infraestructura de seguridad más afianzada, como son los protocolos GSI (Grid Security Infrastructure) basados en clave pública (PKI), utilizados principalmente para la autenticación de usuarios, protección de comunicaciones y esquemas de autorización [6].

3.2 FEDERACION DE IDENTIDAD

Es un tipo de red confiable y conjunto de mecanismos y procedimientos, en donde se tiene como objetivo minimizar la demanda del reconocimiento (autenticación) y validación (autorización) de acceso de los usuarios a los diferentes servicios ofrecidos por los Proveedores de Servicios; Para lograr minimizar estas demandas, se presentó la necesidad de crear y mantener bases de datos con toda la información necesaria sobre los usuarios que pueden acceder a los servicios y su respectivo nivel de privilegio.

El fundamento básico consiste en que toda la información sobre un usuario se encuentra concentrada y mantenida en una única base de datos administrada por la organización a la que se encuentra registrado. Cada usuario debe tener un solo registro (login/password), y cada organización debe definir tanto las políticas de

seguridad para que la información de los usuarios pueda ser mantenida y actualizada, como los métodos de autenticación que se utilizarán; De esta forma se asegura que los proveedores de servicios dispongan sus recursos a los usuarios vinculados a estas organizaciones.

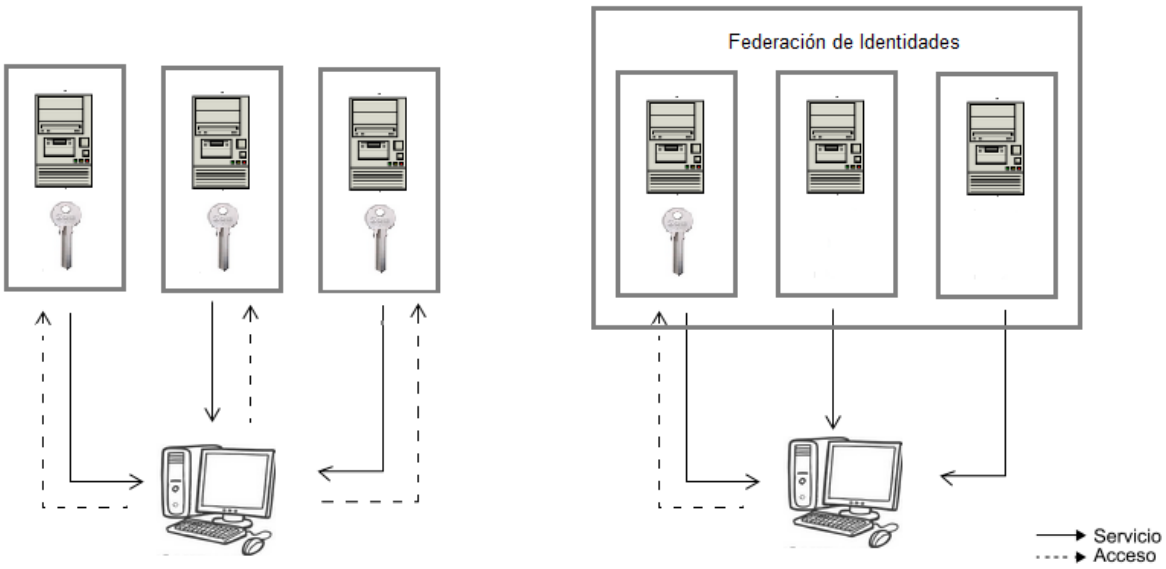

Figura 9. Esquema del Fundamento Básico de Federación de Identidades

Fuente: Autores

Es importante resaltar que los sistemas de identidad federados constituyen, en sí mismos, una relación de confianza entre organizaciones la cual debe estar previamente definida por estas organizaciones, para que la tecnología pueda funcionar sin inconvenientes.

3.2.1 Componentes de una Federación

Una federación está constituida por dos componentes principales, el primero es el Proveedor de Identidad que está encargado de almacenar y gestionar la

información sobre los usuarios y el segundo es el Proveedor de Servicio el cual ofrece los recursos restringidos a grupos de usuarios.

En la arquitectura de una federación, participan tres tipos actores:

- **Usuario**:

Es la persona vinculada a una de las organizaciones y que desea acceder a los servicios restringidos ofrecidos por algún proveedor de servicio perteneciente a la federación.

- **Proveedor del Recurso:**

Son los servicios ofrecidos a los usuarios vinculados a la federación. Estos servicios son asociados al proveedor de servicios incluye tanto los recursos de hardware como software.

- **Institución del Usuario:**

Es la organización que mantiene al proveedor de identidad y establece un proceso interno de autenticación con los usuarios vinculados a ella.

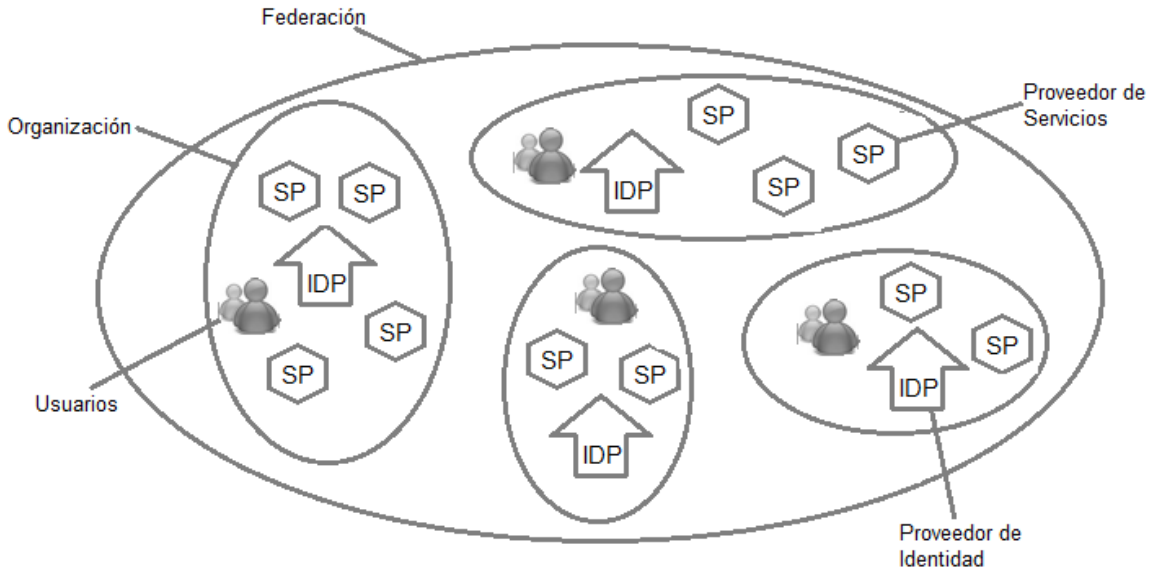

Figura 10. Componentes de una Federación

Fuente: Autores.

Un componente adicional indispensable en una federación, es el Where Are You From (WAYF - ¿De dónde eres?) o Discovery Service (DS), el cual es el componente que se encarga de centralizar las informaciones sobre proveedores de identidad pertenecientes a una federación. Los usuarios de diferentes organizaciones que desean acceder a los recursos ofrecidos por un proveedor de servicio, deben identificar la organización a la cual pertenecen por esta razón se incluye el WAYF a la federación, para ayudar a redireccionar a estos usuarios a su respectivo proveedor de identidad, de esta manera el usuario selecciona la organización con la que está vinculado y comienza a interactuar con su proveedor de identidad para proporcionar sus credenciales.

3.2.2 Proveedor de Identidad

Es una entidad encargada de la política interna de la gestión de identidad de una organización, dicho de otra manera es el responsable de mantener toda la información referente a los usuarios vinculados a la organización.

Solo debe existir un proveedor de identidad por organización para que se encargue de definir los métodos de autenticación interna que se utilizarán, los cuales generalmente consisten en Usuario y Contraseña.

El proveedor de identidad se encarga de proporcionar los atributos y la autenticación del usuario, permitiendo de esta manera que el proveedor de servicio permita a los usuarios el acceso a los recursos o a los servicios.

3.2.3 Proveedor de Servicio

Es una aplicación web que alberga aplicaciones e implementa los servicios y recursos que van a estar disponibles para algunos o todos los usuarios vinculados a una organización, según los privilegios a los que estén autorizados.

Los privilegios se suministran al proveedor de servicio gracias a los atributos obtenidos después de la autenticación realizada por el usuario y la información brindada en el proveedor de identidad, ya que no es función del proveedor de servicios conocer estos atributos e información referente a los usuarios, sino solicitarla al proveedor de identidad.

3.2.4 WAYF / DS

El servicio WAYF (Where Are You From?) Permite a los usuarios seleccionar la

organización a la cual pertenecen, eliminando así la necesidad de que sea el proveedor de servicio el encargado de determinar a cuál de las organizaciones pertenecientes a la federación hace parte un determinado usuario.

Cuando un usuario intenta acceder a un recurso proporcionado por un proveedor de servicio de la federación, se le redirige a la WAYF para que pueda indicar su proveedor de identidad y proceder correctamente con el modelo de autenticación definido por el servidor.

3.2.5 LDAP

Es otro componente adicional en el funcionamiento adecuado de la federación, el cual es consultado por el proveedor de identidad para realizar la autenticación de un usuario. Este es el servidor administrador de directorio, el cual mantiene consolidada la información de las credenciales de autenticación y almacena los atributos de cada usuario de la organización.

Este servidor es como una base de datos, pero en general contiene información más descriptiva y basada en atributos

3.2.6 Interacción entre Componentes de una Federación

Para que un usuario pueda acceder a los recursos de una de las organizaciones que hacen parte de la federación, se realizan una serie de procesos entre el usuario, la organización del usuario y los servicios a los que desea acceder:

- **Paso 1:**

El usuario intenta acceder al proveedor de servicio en donde se encuentran los recursos.

- **Paso 2:**

El proveedor de servicios presenta las opciones de proveedores de identidad que hacen parte de la federación, por medio del WAYF.

- **Paso 3:**

El usuario selecciona de las opciones la organización a la cual pertenece.

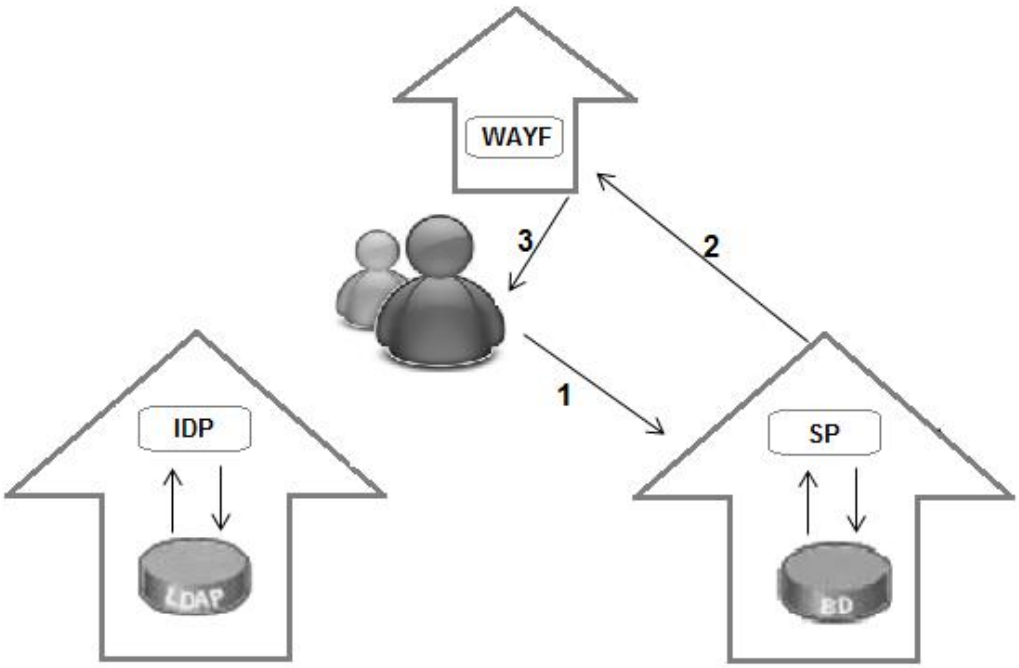

Figura 11. Interacción entre Componentes (Pasos del 1 al 3)

Fuente: Autores.

- **Paso 4:**

El usuario es redirigido a el proveedor de identidad de su organización.

- **Paso 5:**

El proveedor de identidad realiza la autenticación según el método de interno elegido.

- **Paso 6:**

El proveedor de identidad envía al proveedor de servicio la garantía de autenticación del usuario, la cual es un seudónimo que será utilizado únicamente por los servidores para referirse a un usuario en particular.

- **Paso 7:**

Cuando sea necesario, el proveedor de servicios solicita los atributos necesarios adicionales del usuario al proveedor de identidad.

- **Paso 8:**

Con la entrega por parte del proveedor de identidad de los atributos acordados, el proveedor de servicio toma decisiones sobre la base de atributos del usuario y proporciona los servicios al usuario.

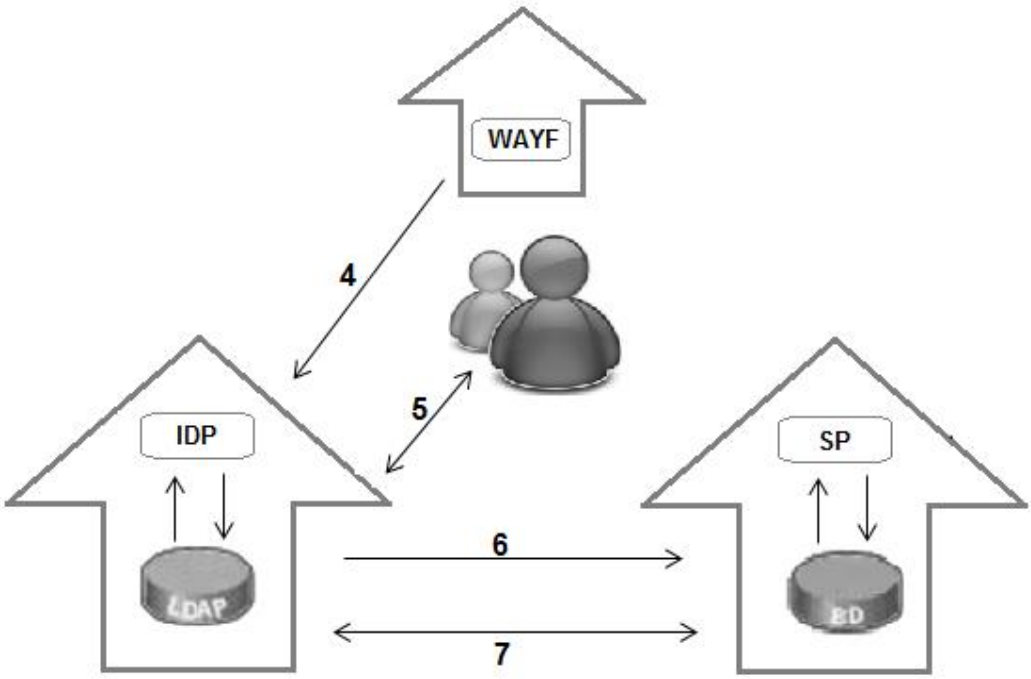

Figura 12. Interacción entre Componentes (Pasos del 4 al 8)

Fuente: Autores.

3.3 UTILIZANDO SSH SOBRE LA FEDERACION DE IDENTIDAD: FEDSSH

FEDSSH es una solución tecnológica basada en OpenSSH que permite integrar servidores SSH con una infraestructura federada obteniendo las claves públicas de sus usuarios desde el LDAP. Su principal característica es que permite el acceso a los recursos no en función de quien es el usuario, sino de los atributos de sus usuarios [10].

Para poder acceder por SSH un usuario tendría que autenticarse en la federación y una vez autenticado, podrá acceder a todos los recursos ofrecidos por la federación sin necesidad de realizar un login nuevamente, basándose en el mecanismo de clave pública, clave privada y estando el servidor en cualquier

entidad de la federación. Esta única contraseña solo se entrega al servidor de la entidad a la que pertenece el usuario.

3.3.1 Proceso de Autenticación

1. El usuario procede a una página específica.

2. El usuario realiza la respectiva autenticación en la federación y puede acceder a la página deseada.

3. La aplicación web (a la que el usuario ingresó) intentará conseguir la clave RSA pública del usuario a través de los datos que son enviados por el servidor de la federación (IDP).

4. Una vez autenticado en la aplicación web, el usuario puede acceder a las maquinas SSH federadas sin necesidad de realizar un nuevo login.

3.3.2 Requisitos

Para realizar la autenticación obteniendo la clave SSH y poder acceder a cualquier maquina remota perteneciente a la federación por SSH, se debe poner el servidor SSH parcheado, este parche de 10 kb es el que permite conectarse al LDAP y obtener la claves públicas.

3.3.3 Caso de Uso

- **Paso 1:**

Antes de poder acceder a cualquier servicio de la federación, el usuario tendrá que autenticarse. Para ello se utilizan los mecanismos que ofrece la federación de identidad. Por tanto el usuario intentará acceder a una aplicación web protegida tras un SP de la federación de identidad.

- **Paso 2:**

Según el funcionamiento de la federación si el usuario no está aún autenticado en la federación de identidad, el SP re-direccionará al usuario hacia el WAYF, dónde seleccionará su entidad de origen, y tras esto será redirigido al IDP de su entidad, donde el usuario proporcionará sus credenciales.

- **Paso 3:**

Una vez autenticado en la federación de identidad el sistema re-direccionará al usuario al SP, pasándole a este los datos necesarios para asegurar la identidad del usuario.

- **Paso 4:**

La aplicación principal, tras el SP, recibirá entonces los datos necesarios del usuario que le fueron proporcionados por el IDP. Y con estos datos creará una entrada en el servidor de claves, el cuál será consultado por los servidores SSH para verificar que el usuario esté autenticado en la federación. En este paso es

donde se comienza a sacar la federación de identidad del ámbito Web.

- **Paso 5:**

En este punto el usuario ya está autenticado y ahora puede entrar en uno o varios servidores SSH federados sin necesidad de escribir su contraseña.

- **Paso 6:**

El servidor SSH tiene que saber si el usuario que intenta acceder está autenticado en la federación por lo que consultará el servidor de claves para confirmar que dicho usuario está autenticado y además es quien dice ser.

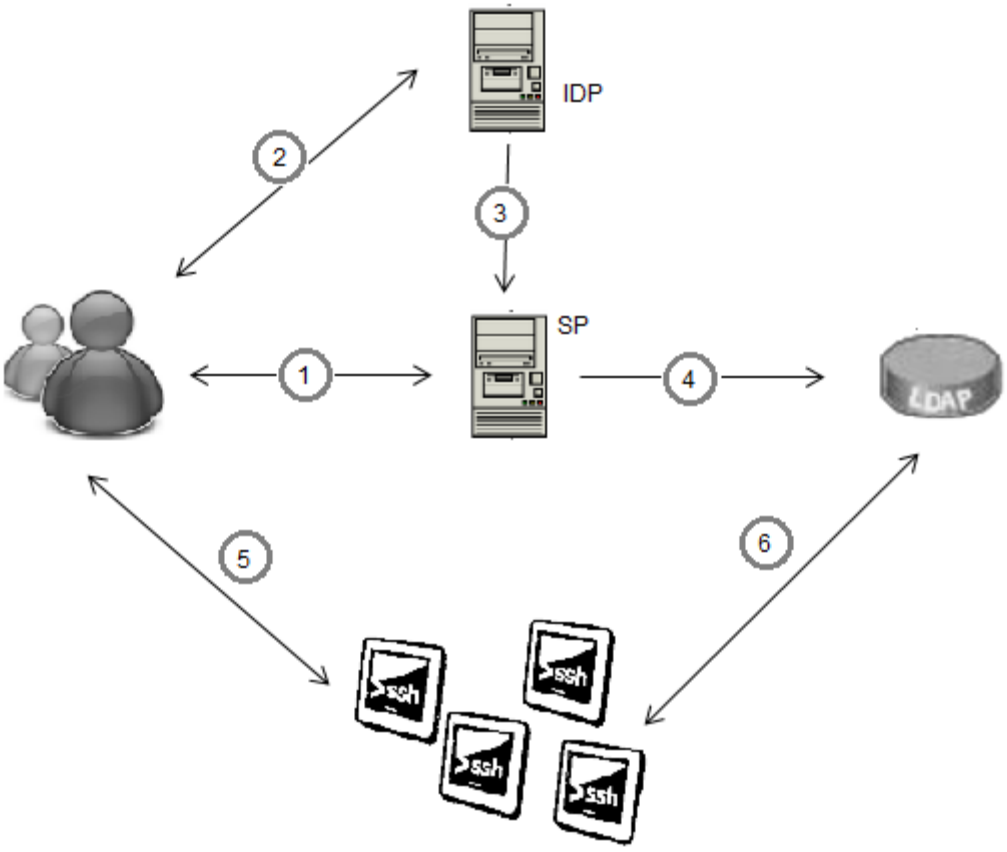

Figura 13. Caso de Uso

Fuente: Autores.

4. ESTADO DEL ARTE

La federación de identidad se basa en el intercambio de información de sus usuarios entre diferentes dominios de identidad o proveedores de servicios, para lo cual es necesario disponer de un estándar propicio que permita el manejo de un lenguaje común entre los sistemas.

Como respuesta a las necesidades que han surgido con relación a la federación de identidades, recientemente han nacido diversas iniciativas originadas generalmente por asociaciones o uniones empresariales; que han arrojado como resultados diversos estándares, a continuación se mencionan aquellos de mayor relevancia e impacto en el ámbito académico y empresarial [12]:

1. Microsoft/IBM
2. OASIS
3. Liberty Alliance
4. PAPI

4.1 MICROSOFT/IBM

Para el 2004, IBM y Microsoft crean un protocolo para la creación de Web Service seguros. Esta especificación se llamó WS-* (ahora es llamado WSS) y está basado en la creación de tokens que se adjuntan a los mensajes, algunos de estos tienen que ver con la identidad.

Este protocolo se estructura de forma modular, aislando componentes de la seguridad en diferentes especificaciones. A Continuación describimos los más

importantes respecto a identificación:

- **WS-Policy:**

Es la política de seguridad de un determinado web service, mediante un lenguaje de programación bien sea SAML o Kerberos.

- **WS-Trust.:**

Consiste en un lenguaje que permite a los servicios de una autoridad poder intercambiar token con otra en la que se confía.

- **WS-Federation:**

Es un lenguaje que organiza las interacciones de los objetos WS-Trust (identidades, atributos, autenticaciones, etc.) participantes en un servicio web.

4.2 OASIS

SAML (Security Assertion Markup Language) de OASIS (Organización para el Desarrollo de Estándares de Información Estructurada) es un consorcio sin fines de lucro que impulsa el desarrollo, convergencia y adopción de estándares abiertos para la sociedad de la información global. OASIS promueve consenso de la industria y produce normas internacionales para la seguridad, el cloud computing, SOA, servicios web, las redes inteligentes, la publicación electrónica, gestión de emergencias, y otros ámbitos.

La esencia de SAML se basa que un cliente pide algo sobre una identidad a una autoridad, que a su vez responde con aserciones. De la cual se puede tener 3

tipos de aserciones.

- **Aserción de Autenticación:**

El sujeto de la aserción fue autenticado por medios particulares en un tiempo concreto. Estas aserciones son emitidas por la parte que identificó con éxito al usuario. Definen quien emitió la aserción, el sujeto autenticado, el periodo de validez, además de otras informaciones relacionadas con la autenticación.

- **Aserción de Atributo:**

El sujeto de la aserción se asocia a los atributos proporcionados, de esta forma toda entidad que tenga establecida una relación de confianza con la autoridad SAML, tendrá la certeza de que el sujeto tiene ciertas características.

- **Aserción de Autorización:**

Una petición para permitir al sujeto de la aserción acceder al recurso solicitado.

Generalmente una aserción contiene:

1. ID del emisor y fecha de expedición.
2. ID de la aserción.
3. Sujeto, nombre del dominio y opcionalmente los datos de autenticación.
4. Información adicional de la entidad emisora.
5. Condiciones bajo la cual la aserción es válida.
6. Restricción de audiencia.

7. Restricción de destino (ej: la URL).

8. Condiciones específicas de la aplicación.

4.3 LIBERTY ALLIANCE

Liberty Alliance[11] es un consorcio de proveedores de tecnología y empresas orientadas al consumidor formados para desarrollar una norma abierta para la identificación de red federada. El proyecto Liberty Alliance se basa en el concepto de que los usuarios puedan conectar varios conjuntos de datos personales que existen en distintos proveedores de comercio electrónico en una sola identidad federada, fácil de manejar. Esto permite la comodidad de un mecanismo de SSO así como la administración más fácil de la información personal a través de múltiples proveedores de servicios. Liberty Alliance es una de las más destacadas propuestas de estándares de identidad federada contando con tres especificaciones diferentes. Entre ellas están:

1. Identity Federation Framework (ID-FF)

Permite el SSO a los consumidores y usuarios de servicios en Internet y aplicaciones de comercio electrónico, visitar o participar en los servicios de varios sitios web que tienen una relación de confianza. Este enfoque federado no requiere que el usuario vuelva a autenticarse y puede soportar controles de privacidad establecidos por el usuario.

2. Identity Web Service Framework (ID-WSF)

Es un Framework para implementar y administrar una variedad de servicios de identidad en Web. Desarrollado sobre la base de los requerimientos del negocio

bien definidos y con controles para los consumidores y la privacidad del usuario, aplicaciones Liberty Web Services incluyen Ubicación geográfica, libreta de contactos, calendario, mensajería móvil y Libertad People Service, primer Framework de servicios web abiertos de la industria para la gestión de aplicaciones sociales tales como marcadores, blogs, calendarios, intercambio de fotos y mensajería instantánea en una red social federada segura y respetando la privacidad.

3. Identity Services Interface Specifications (ID-SIS)

Se utiliza para la construcción de un conjunto de servicios (ubicación geográfica, libreta de contactos, calendario, mensajería móvil) interoperables sobre ID-WSF.

4.4 PAPI

Es un sistema federación desarrollado por RedIRIS desde el año 2001. RedIRIS es una red académica y de investigación española, proporciona servicios avanzados de comunicaciones a la comunidad científica y universitaria nacional. Cuenta con más de 450 instituciones afiliadas, principalmente universidades y centros públicos de investigación, que llegan a formar parte de esta comunidad.

El funcionamiento de PAPI es bastante similar al SSO de Liberty/SAML 2.0: la aserción es transmitida mediante el método POST de HTTP desde un proveedor de autenticación hasta los proveedores de servicio, sólo que dicha aserción no es en XML.

Los componentes principales del sistema son: El servidor de autenticación (AS) y

los puntos de acceso (PoA). Un PoA se encarga de interceptar los accesos a recursos o servicios forzando la autenticación del usuario en un AS (cada PoA puede estar conectado a uno o varios AS),La autenticación se produce contra alguno de los backends configurados en el AS (Ldap, BD, etc) y tras producirse se genera una aserción que se transmite al POA, y que contiene datos del usuario autenticado.

Una de las federaciones más activas en lo que se refiere a investigación y desarrollo de aplicaciones para la federación de identidad es la federación de Noruega (Feide). El proyecto de SSH sobre federación de identidad nace a partir de un documento publicado el 21 de agosto del 2007 por Feide. Este documento investiga cómo las credenciales de federación de identidad pueden ser usadas para autenticar a diferentes servicios como el SSH. A Partir de esta idea dando un paso más y tratando de automatizar al máximo el proceso RedIRIS desarrolla Fedssh una aplicación web que federa SSH, dando mayor comodidad al usuario y al administrador.

5. METODOLOGÍA

El proceso metodológico a seguir es una adaptación del método científico, a través de este se va una concepción y diseño del sistema. Las etapas se organizan así:

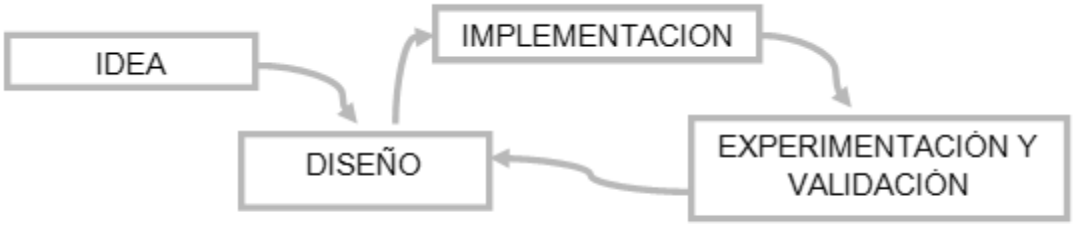

Figura 14. Proceso Metodológico
Fuente: Autores.

El proyecto se realizará en 4 fases cada una de la mano de la otra y en constante realimentación.

Fase 1: Análisis y Diseño del Esquema de Seguridad

Esta etapa inició con el proceso de documentación e investigación acerca de trabajos realizados en el área de supercomputación y federación de identidad, en conjunto o por separado. La información recopilada permitió consolidar una estrategia, logros y aportes de una federación de identidad tanto en nivel académico como comercial. En esta fase, se realizó consultas, que permitió a los autores conocer el proceso e implementación de diversas alternativas de federación de identidad, enriqueciéndose con cada una de las propuestas.

Además de esto se pudo establecer las ventajas y desventajas de cada protocolo.

La literatura para obtener esta información fue principalmente tomada de las páginas web de cada uno de los consorcios de cada protocolo de federación como también de bases de datos reconocidas como lo es la IEEE y ACM. En esta últimas se encuentran un conjunto de artículos enmarcados en el tópico de Comunicación, difusión y creación de redes, cómputo y procesamiento (Hardware y Software) e ingeniería.

En base a lo encontrado, se inicia la segunda etapa de esta misma fase lo cual se basó en la selección de información puntual sobre federación y federación sobre SSH, ya que lo que se quiere federar con este prototipo es Guane el cual trabaja bajo este protocolo, también se seleccionó información sobre esquemas y métodos para la implementación de una federación de identidad. Durante la selección se hizo un proceso de filtrado de la información, debido a que es un tópico muy general con gran volumen de trabajos realizado a nivel internacional, que aunque tratan el tema de federación de identidad, la mayoría no van enfocados a federar HPCaaS.

Resultados alcanzados:

1. Se catalogaron los artículos afines a este proyecto seleccionados de base de datos internacionales y links de documentación de desarrolladores de tecnologías para federar identidad.

2. Se obtuvo la primera definición por parte de los autores acerca del concepto de federación de identidad.

3. Reconocimiento de vocabulario especializado relacionado con la supercomputación y la federación de identidad.

4. En base a los conocimientos adquiridos, se realizaron diseños de los primeros esquemas de seguridad, el cual uno de estos representaría la base para la elaboración de ésta tesis.

Dificultades:

Uno de los inconvenientes encontrados fue la falta de literatura física como libros o revistas científicas que presentarán el desarrollo completo de este trabajo a nivel institucional. La biblioteca de la UIS no tiene trabajos anteriores en este campo.

La falta de documentación también se presenta a nivel latinoamericano, aclarando que la federación CAFE implementada en Brasil, solo presta servicios a nivel de recursos web y no HPCaaS, esto sustentado en que la mayor parte de trabajos reconocidos que se encontraron en la web son de universidades, empresas y centros de investigación de países como: Estados Unidos, Francia y Noruega.

Fase 2: Determinar la Estrategia Tecnológica

Etapa base del proyecto, donde se determinó los recursos tecnológicos requeridos por el esquema de seguridad diseñado e implementación del prototipo.

Resultados alcanzados:

1. De los posibles esquemas diseñados en la fase anterior, se escogió el esquema de federación de identidad a HCPaaS mediante federación de SSH, como esquema a desarrollar.

2. Se estableció el listado de los recursos tecnológicos necesarios para montar el esquema seleccionado. Se realizó una evaluación de los recursos de hardware con que contaba la universidad. Una vez realizada la evaluación, se definió la estrategia tecnológica que cumplirían los objetivos del proyecto, abarcando los tres aspectos necesarios para conseguir la interacción segura, para usuarios-plataformas se escogió SSH, usuarios-aplicaciones se definió un navegador web con interacción segura de protocolo HTTPS, accersiones seguras y comportamiento de atributos esenciales y no tan sensibles del usuario y entre aplicaciones-plataformas se escogió SimpleSAMLphp por parte del proveedor de servicios y Shibboleth en el proveedor de identidad, utilizando protocolos OpenSSL para la interacción segura entre servidores (Shibboleth y SimpleSAMLphp) y entre Proveedor de identidad y usuario.

3. Se inicia con la selección del sistema operativo. Si bien existía una claridad de uso de software libre, escoger uno para cada servidor entre la cantidad de distribuciones existentes fue la primera decisión a tomar por los autores.
Basados en la distribución con mayor documentación en cuanto a sistema operativo como de federación de identidad, se seleccionó el sistema operativo Ubuntu 12.04 basado en un núcleo Linux.

4. Una vez definido el esquema, tecnologías y sistemas operativos a utilizar, se realizó la implementación, lo primero que se realizó fue la configuración de un

LDAP de prueba donde se incluyeron usuarios y contraseñas para realizar pruebas, una vez alcanzado este nivel se configuro el servidor IDP y se probó la conexión con el LDAP de prueba, luego se comenzó con la configuración de Shibboleth en el proveedor de servicio dando fin a la implementación del primer prototipo.

Dificultades:

Una dificultad, surgió en la configuración del LDAP, en la definición de políticas para él envió de atributos de parte del proveedor de identidad hacia el proveedor de servicios.

La principal dificultad en esta etapa, se presentó en la implementación del Proveedor de Servicio, ya que para esta configuración se necesita una amplia documentación la cual es muy limitada en cantidad.

Fase 3: Evaluación del Prototipo

Esta etapa se lleva a cabo en la mayor parte del desarrollo del proyecto, dado el caso que los resultados no cumplan por completo las expectativas, se realizó una constante retroalimentación en el comportamiento del prototipo y la generación de ajustes para ofrecer una solución al esquema de seguridad.

Resultados alcanzados:

1. Al terminar la configuración del LDAP de prueba e implementación del IDP se realizó una serie de pruebas para asegurar la conexión entre ellos y si cumplía los

objetivos planteados. Se observó que se cumplían las expectativas.

2. Cuando se terminó la implementación del SP, se notó que hasta ese punto el prototipo solo servía para federar servicios web, lo cual no cumplía con los objetivos planteados. De esta manera se empezó un nuevo ciclo de análisis, para modificar el esquema planteado, se encontró que configurar la aplicación web FedSSH en el SP y cambiar el sistema a SimpleSAMLphp era la mejor alternativa para cumplir con el objetivo trazado, ya que FedSSH es una aplicación web que integra SSH y Federación, la cual nos permitiría federar HCPaaS en la universidad.

Dificultades:

Como las nuevas configuraciones se le realizan al SP, la falta de documentación presentó nuevamente un gran impedimento en el avance de esta etapa.

Se encontraron otras dificultades: como en el momento de realizar las pruebas al servidor de identidad con una organización externa; Adaptar el LDAP al nuevo esquema de federación de SSH e instalar la aplicación FedSSH en la maquina ya que posee poca documentación.

Fase 4: Divulgación de los Resultados

En esta última fase se dio a conocer el esquema obtenido durante todo e l proceso y se realizó su respectiva documentación.

Resultados alcanzados:

1. Una vez terminada cada fase del proyecto, se realiza la respectiva documentación, a la cual se la realizó una constante realimentación por parte de los autores.

2. Una vez terminado el proyecto y verificando que cumplía con los objetivos planteados y después de las respectivas pruebas, se realizó una reunión con el grupo de investigación de Supercomputación y Cálculo científico junto con los profesores, en la que por medio de una presentación de dio a conocer el proyecto de principio a final.

6. IMPLEMENTACIÓN

La implementación del proyecto se divide en cuatro etapas, las cuales cada una de ellas determinan de manera significativa el desarrollo y éxito de su consecuente, estas son: Especificación y definición del esquema, definición del hardware, definición del software, manejo del software y resultado de la implementación del prototipo.

6.1 ESPECIFICACIÓN Y DEFINICIÓN DEL ESQUEMA

Cuando se desea realizar la implementación de una federación de identidad, por definición se sabe que tienen dos servidores como pilares de su funcionamiento, el Proveedor de Servicios y el Proveedor de Identidad.

El proveedor de servicios, es el servidor que contiene a la aplicación web, ya sea a la que se desea ingresar (cloud) o una aplicación que cumple una funcionalidad particular a la federación, sea cual sea su funcionalidad, esta aplicación web es la encargada de redirigir al usuario al proveedor de identidad al que pertenece para realizar la respectiva autenticación.

El proveedor de identidad, es el servidor que provee la autenticación del usuario y devuelve los datos del mismo que el proveedor de servicios requiere para autorizar el acceso a los servicios deseados. Constantemente existe un proveedor de identidad por organización y como se sabe, una federación está constituida por varias organizaciones, por lo tanto es necesario que el usuario seleccione la entidad a la cual pertenece y para realizar ante esta su identificación. A este proceso de identificar al proveedor de identidad se conoce como WAYF. El

proveedor de identidad está complementado por un componente adicional, al cual este consulta para obtener la información descriptiva del usuario basada en atributos.

La siguiente figura es la representación del primer diseño que se implementó para la culminación de esta tesis, En este esquema, se representa una federación de identidad en donde los servicios prestados a los usuarios solo son en web:

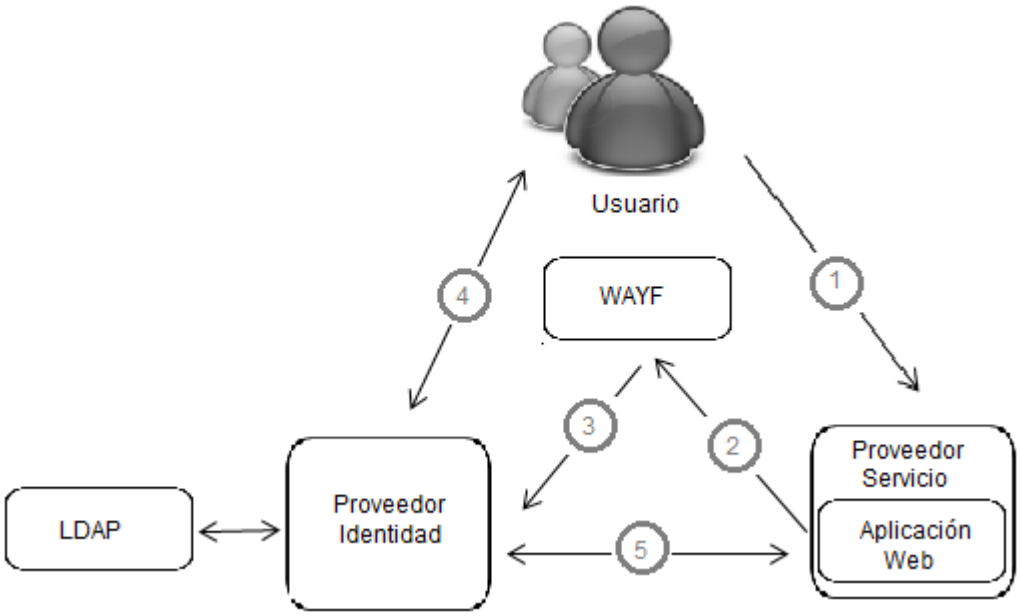

Figura 15. Representación del Primer Esquema

Fuente: Autores.

En este primer prototipo tanto el proveedor de identidad como el de servicios tienen implementados los estándares de Shibboleth utilizados para la autenticación y autorización federea vía web.

Al realizar la evaluación de este esquema, se notó que no se cumplían con los objetivos propuestos, ya que efectivamente se implementaba un prototipo de

federación, pero este ofrecía servicios que no cumplían con las expectativas de esta tesis, por esta razón fue necesario una realimentación en donde se diseñó e implementó un nuevo esquema, el cual surgió de las complementaciones realizadas al esquema anterior.

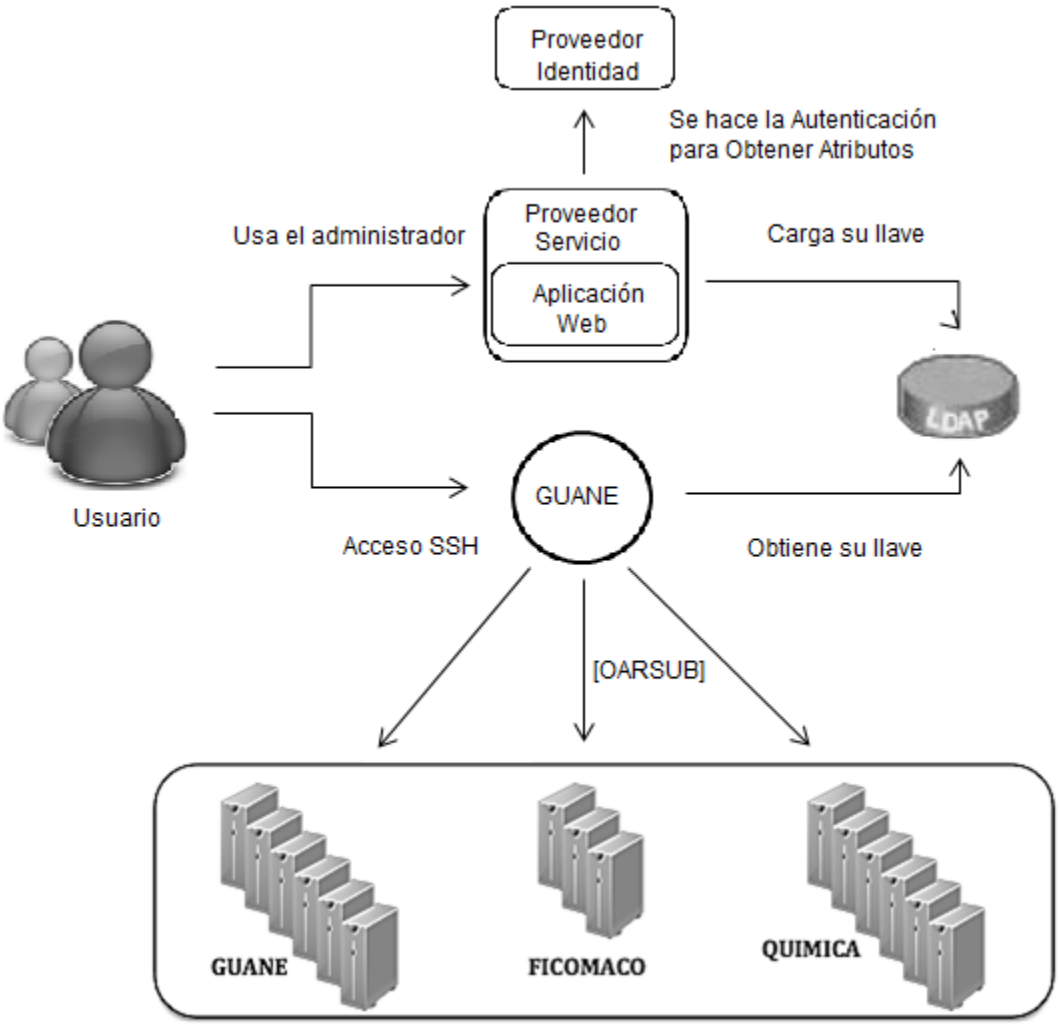

Figura 16. Esquema Final

Fuente: Autores.

Para lograr el objetivo propuesto prestando los servicios establecidos (HPCaaS),

en este esquema se agregó al funcionamiento del esquema anterior FedSSH, la cual es una aplicación web que se encuentra incorporada en el Proveedor de servicios.

En esta nueva configuración el proveedor de servicios tiene instalado el protocolo SimpleSAMLphp como base y es complementado con la aplicación gestor de certificados (FedSSH); Cuando se ingresa por primera vez a una de las plataformas ofrecidas como servicios por la federación, es necesario crear un certificado en el LDAP, este certificado es generado por el usuario y es este el encargado de subirlo mediante la aplicación web, quien es la encargada de almacenarlo en el LDAP, si es necesario actualizar la base con un nuevo certificado se realizará por medio de esta aplicación web.

Como tal, el comportamiento del esquema en comparación con el anterior y sabiendo que cumple con las mismas bases, es muy similar aunque con diferentes tecnologías; Ahora, el usuario aunque no forme parte de la organización teniendo sus certificados previamente generados en su máquina, entra a la página FedSSH y guarda sus atributos en el LDAP incluyendo su certificado, después de esto el servidor SSH federado (plataforma) podrá dar acceso a los usuarios, este verifica si el usuario es quien dice ser y tiene acceso a los recursos, consultando en el LDAP y obteniendo las claves publicas entre atributos del usuario

6.2 DEFINICIÓN DEL HARDWARE

La implementación de este prototipo se llevó a cabo con componentes y equipos proporcionados por la Universidad Industrial de Santander (UIS) y la unidad de supercomputación y calculo científico SC3 de la misma. El sistema se implementó con dos servidores de características estándares.

Las especificaciones de hardware de los 2 servidores que hacen parte de este prototipo, se describen en la siguiente tabla:

	Servidor 1 **Proveedor de Identidad**	**Servidor 2** **Proveedor de Servicios**
Procesador	Intel® Pentium(R) 4 CPU 3.20GHz × 2	Intel® Pentium(R) 4 CPU 3.20GHz × 2
Memoria RAM	2,0 GB	512 MB
Disco Duro	76,4 Gb	155,4 Gb

Tabla 1: Especificaciones de Hardware

6.3 DEFINICION DEL SOFTWARE

La definición del software dentro del abanico existente para la implementación de una federación de identidad, puede variar dependiendo de los conocimientos que se tengan al momento de comenzar con la implementación del prototipo. En este libro se describe el software implementado en cada servidor y las razones por las cuales se seleccionó esa herramienta en particular.

1. Sistema Operativo

Para la implementación de esta tesis, se eligió como sistema operativo de los dos servidores a Ubuntu 12.04 LTS, debido a que es un sistema operativo estable y es el contiene un gran número de documentación.

2. Software del Proveedor de identidad

Shibboleth 2.3.8, fue la herramienta seleccionada para la configuración de este servidor, gracias a su documentación, ya que son muchas las organizaciones que tienen en su implementación esta herramienta gracias a la interoperabilidad que posee.

Figura 17. Proveedor de Identidad del prototipo implementado

Fuente: Autores.

Además de su amplia documentación, Shibboleth es una de las soluciones de identidad en el mundo de mayor despliegue federado y se encuentra bajo la licencia de apache en software libre.

El funcionamiento de la autenticación y entrega de atributos del proveedor de identidad de shibboleth, se lleva a cabo de la siguiente manera:

1. El usuario envía sus credenciales que son debidamente verificados por el idp.

2. El proveedor de identidad envía un hangle (identificador) al proveedor de servicios certificando que el usuario ha sido autenticado

3. El proveedor de servicios envía un hangle al proveedor de identidad, solicitando la entrega de los atributos.

4. El proveedor de identidad luego de consultar al LDAP entrega los atributos al proveedor de servicios.

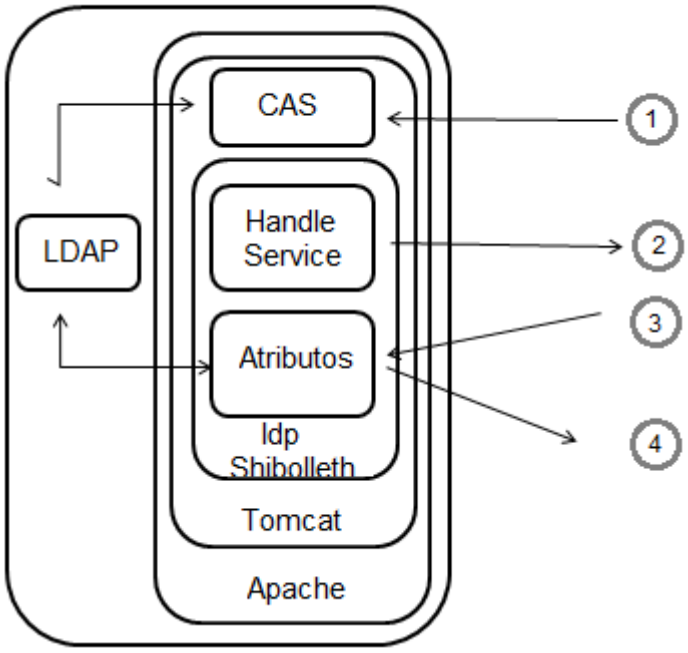

Figura 18. Proveedor de identidad de Shibboleth

Fuente: Autores.

3. Software del Proveedor de Servicios

El software instalado en este servidor es el simpleSAMLphp, el cual contiene y

protege de acceso no autorizado a la aplicación FedSSH, fundamental para el comportamiento de este esquema.

Figura 19. Proveedor de Servicios del Prototipo Implementado

Fuente: Autores.

SimpleSAML, es una herramienta de la empresa UNINETT basado en estándares, es muy flexible ya que posee un sistema de módulos y de filtros que permite añadir fácilmente una funcionalidad extra, en nuestro caso esa funcionalidad extra es la aplicación FedSSH.

El funcionamiento de SimpleSAML, como proveedor de servicios se puede ver en la siguiente figura:

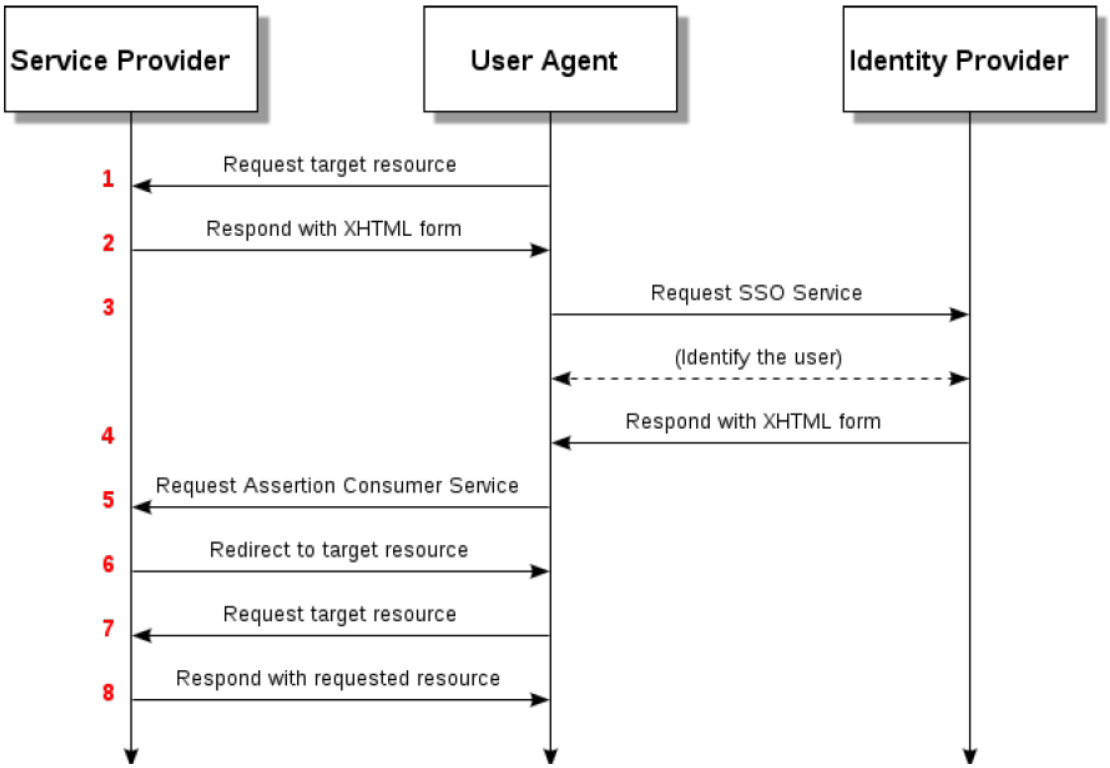

Figura 20. Funcionamiento SimpleSAML

Fuente: Autores.

3. Esquemas y Atributos Utilizados en el LDAP

Los sistemas de directorio utilizan diferentes atributos para guardar los datos pertenecientes a los diferentes objetos. Cada posible atributo a utilizar está definido en un “Esquema LDAP", y se podrá utilizar siempre que el objeto tenga el “objectclass" que define ese atributo.

Así pues, es necesario definir los atributos que se necesitan para almacenar los usuarios junto a sus claves en el servidor de claves (LDAP).

Los atributos utilizados en el prototipo de esta tesis, son los siguientes:

1. uid: del usuario autenticado en la federación.

2. clave pública del usuario autenticado en la federación.

3. timeout tiempo de vencimiento de la sesión del usuario autenticado en la federación.

4. edupersonaffiliation usuarios, profesores y más. Necesario para federaciones internacionales

La siguiente tabla muestra los esquemas utilizados para cada atributo:

ATRIBUTOS	ESQUEMAS
Uid	person
Clave pública	ssh-lpk[5]
Timeout	schac[6]
edupersonaffiliation	eduPerson[7]

Tabla 2: Esquemas

6.4 RESULTADO DE LA IMPLEMENTACION DEL PROTOTIPO

Al final de toda la implementación se tiene un prototipo de Federación de Identidad que como servicio se ofrece HCP.

[5] http://dev.inversepath.com/openssh-lpk
[6] http://www.terena.org/activities/tf-emc2/schac.html
[7] https://www.feide.no/attribute/edupersonaffiliation

En la siguiente tabla se observan las direcciones IP de cada uno de los componentes del prototipo después de la implementación:

	URL
PROVEEDOR DE IDENTIDAD	https://192.168.66.251/idp/Authn/UserPassword
WAYF / SP	https://192.168.66.252/simplesamlphp
FEDSSH HOME	https://192.168.66.252/fedssh/
FEDSSH SERVER	https://192.168.66.252/fedssh/server.php
FEDSSH KEYS	https://192.168.66.252/fedssh/Keys.php

Tabla 3: Direcciones IP del Prototipo

7. PRUEBAS Y ANALISIS DE RESULTADOS

Se hicieron dos pruebas al prototipo en la cual ambas se fundamentaron en la autenticación y autorización del esquema. Este punto se dividió en dos partes localmente y exteriormente.

7.1 PRUEBA LOCALMENTE Y DE LA APLICACIÓN FEDSSH

En esta prueba se tuvo en cuenta la autenticación y los atributos enviados al proveedor de servicios por parte del proveedor de identidad. Se obtuvo los siguientes resultados descritos en la imagen de la aplicación FedSSH.

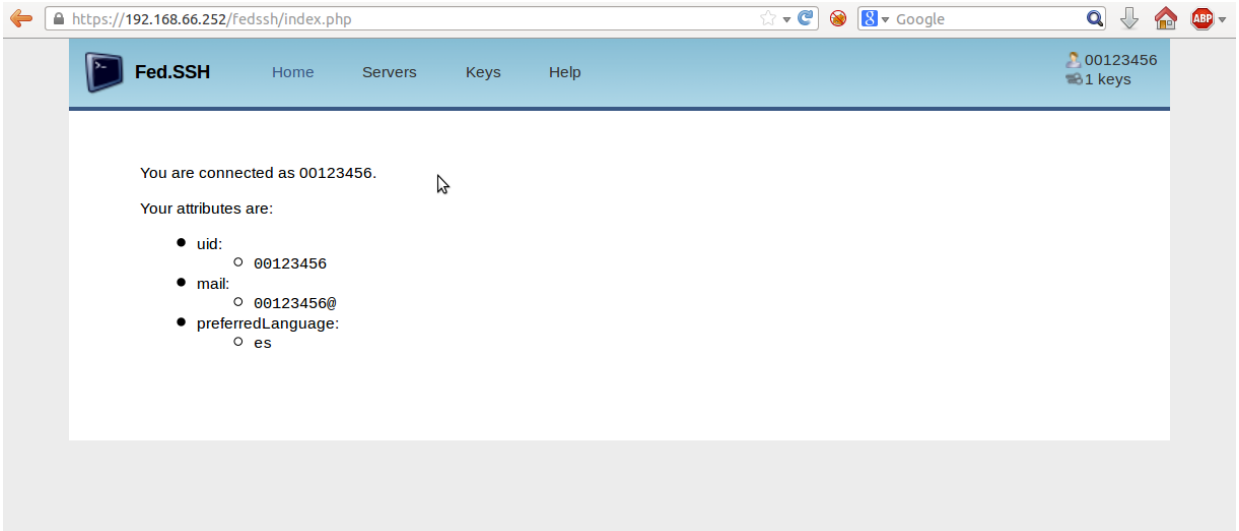

Figura 21. Resultado Prueba localmente y Aplicación FedSSH

Fuente: Autores.

Se pueden observar los atributos enviados por el IDP mediante la previa autenticación del usuario.

7.2 PRUEBA EXTERIOR TESTSHIB

La siguiente prueba que se hizo fue la posibilidad de gestionar identidad fuera de los recursos de la Universidad Industrial de Santander y poder tener acceso a otros recursos. Esta prueba se hizo también en ámbito de prueba debido a que solo se está trabajando un prototipo. La prueba se generó en TestShib una aplicación web de Shibboleth que permite ver si el proveedor de identidad está correctamente configurado y ver que atributos está enviando el servidor. De la prueba se obtuvo el resultado descrito en la siguiente imagen.

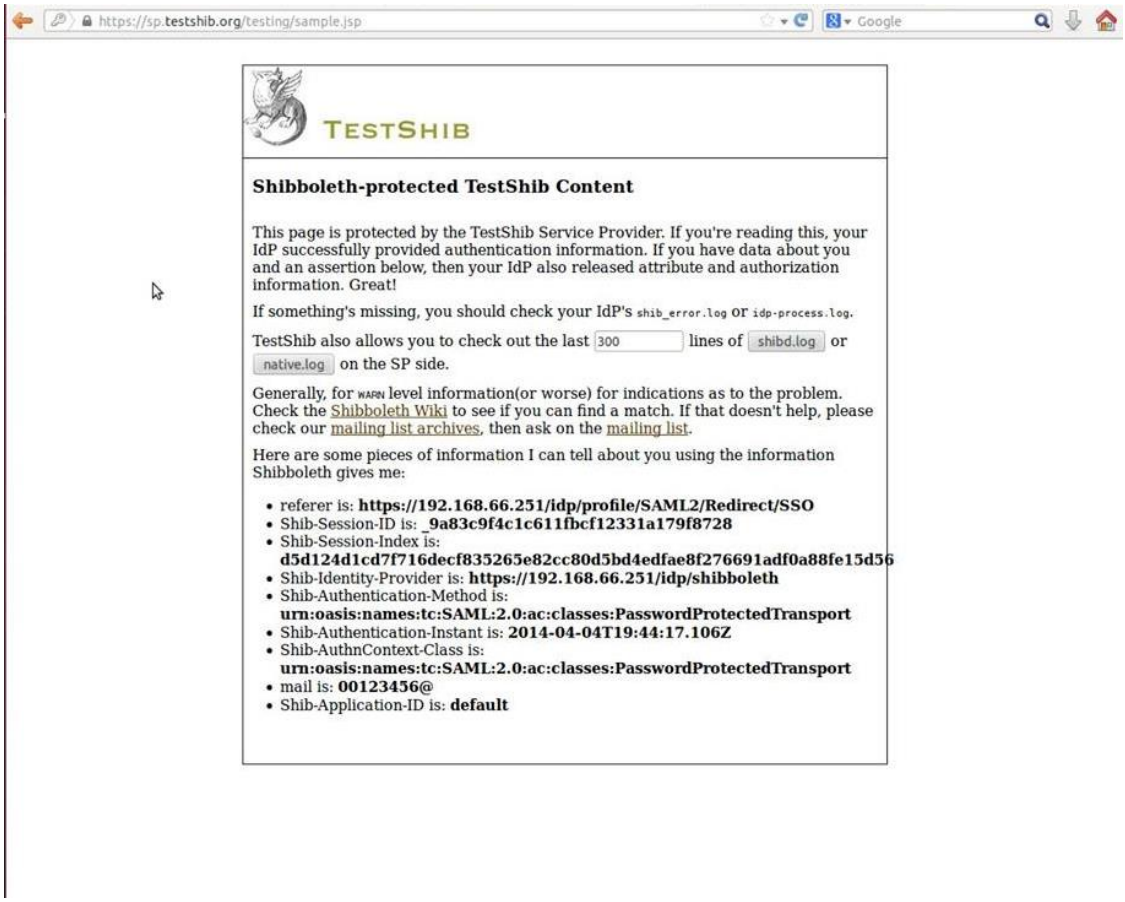

Figura 22. Resultado Prueba Exterior TESTSHIB

Fuente: Autores.

Se escogió esta herramienta ya que permite probar el servidor de forma automáticamente, ágil y eficaz en la plataforma. Permitiendo obtener resultados rápido y contundentes acerca de la prueba.

8. CONCLUSIONES

Este proyecto consiste en la unión de diferentes herramientas que funcionan en conjunto para formar un prototipo de federación de identidad de tal manera que ofrezcan un servicio de HCP. Al principio iniciar con el prototipo puede ser algo tedioso, puesto que en caso de esta tesis hay muchas herramientas involucradas y la federación de identidad es todavía una tecnología poco documentada.

La infraestructura que se obtuvo para proporcionar el control de gestión de identidades y accesos tiene como objetivos: autenticar los servicios de nube utilizando las políticas de privacidad de los usuarios, proporcionando una información mínima al proveedor de servicios, garantizar la protección mutua de clientes y proveedores. En este trabajo se destaca el uso de herramientas específicas, Shibboleth en el proveedor de identidad, que brinda apoyo a las tareas de autenticación, autorización y federación de identidades. SimpleSAMLphp conjunto con FedSSH brinda el servicio de federación del protocolo SSH como Proveedor de Servicios.

Shibboleth es muy flexible en cuanto a su uso en un entorno de nube, lo que permite a un servicio prestarse de forma fiable y segura. Además, Shibboleth se basa en SAML, lo que significa que es compatible con las normas internacionales, lo que garantiza la interoperabilidad. Vistas en este proyecto donde se utiliza diferentes aplicaciones de federación.

Se puede concluir de esta experiencia, que las licencias libres ayudan a la innovación y aprendizaje ya que se puede basar en el código y conocimiento de

otras personas para realizar prestaciones de servicios de diferentes índoles como es HPCaaS. Por otra parte, cada vez se está imponiendo más el concepto de la federación de identidad, tanto en el ámbito universitario como en el empresarial, por lo tanto el desarrollo de esta tesis puede ayudar al crecimiento de grandes proyectos a desarrollar en el futuro.

Este prototipo hace que la prestación de servicios de HPC fuera de la universidad se simplifique al utilizar los conceptos de la federación de identidad en el acceso por SSH, así como la federación de identidad simplifica la gestión de los usuarios a las aplicaciones web.

Al analizar y realizar la evaluación del prototipo implementado en la interacción tanto de servicios privados como públicos, se puede concluir que se alcanzaron los objetivos propuestos al inicio de esta tesis, ya que, se propuso un esquema de seguridad para HPCaaS, a partir del esquema base de federación de identidad, sin alterar los lineamientos de autenticación y federación de identidad.

El análisis del esquema se demostró que el acceso a los recursos de cómputo son seguros, ya que se realizan a través de los protocolos HTTPS y SSH, los cuales realizan encriptaciones de tráfico de datos. Como también permite interoperabilidad con otros recursos web federados.

Se definieron las estrategias tecnológicas (SimpleSAMLphp, Shibboleth y FedSSH) para la interacción entre los distintos componentes que forman parte de la federación.

9. RECOMENDACIONES

Para futuros desarrollos del prototipo, se recomienda tener experiencia en el manejo de herramientas de federación. Se recomienda tener en cuenta la disponibilidad de librerías y paquetes en los diferentes sistemas operativos que pueden ser un inconveniente a la hora de la implementación.

Para la implementación de una federación de identidad se recomienda tener en cuenta los certificados de seguridad firmados por autoridades encargadas de generar estas firmas dándole confianza a los usuarios y a las organizaciones que pertenezcan a la federación.

Es recomendable elaborar un portafolio de documentos previo a la implementación del Proveedor de Servicio, ya que la principal dificultad en esta etapa es la falta de documentación.

10. LIMITACIONES

La principal limitante, es la dificultad de plantear aplicaciones de uso generales sobre la federación, ya que al ser un recurso definido por políticas de usuarios específicas y las aplicaciones que pueden hacer uso de una federación son numerosas, cada una de ellas con unas especificaciones y necesidades propias. En base a esto, el personal encargado de la administración de estos recursos debe estar en la capacidad de brindar soluciones casi personalizadas a cada servicio.

La segunda limitante durante el proyecto, fue la poca documentación que había para la implementación de un esquema de federación de identidad para HPCaaS, aunque existen varios artículos sobre la historia, arquitecturas, avances, etc., sobre federación de identidad, no existe una amplia documentación acerca del uso de federación de identidad para HPCaaS. Debido a que es reciente este nuevo esquema y pocas Organizaciones o Universidades lo han abordado.

REFERENCIAS BIBLIOGRAFÍCAS

[1] “HPC: It’s Not Just for Rocket Scientists Any More”. En: PhD Eadline, Douglas. *High Performance Computing For Dummies®, Sun and AMD Special Edition.* Indianapolis, Indiana: Wiley Publishing, Inc., Copyright 2009. pp. 3

[2] “Getting to HPC”. En: PhD Eadline, Douglas. *High Performance Computing For Dummies®, Sun and AMD Special Edition.* Indianapolis, Indiana: Wiley Publishing, Inc., Copyright 2009. pp. 9

[3] “Categories of Clusters”. En: PhD Eadline, Douglas. *High Performance Computing For Dummies®, Sun and AMD Special Edition.* Indianapolis, Indiana: Wiley Publishing, Inc., Copyright 2009. pp. 13

[4] “If You Need Speed”. En: PhD Eadline, Douglas. *High Performance Computing For Dummies®, Sun and AMD Special Edition.* Indianapolis, Indiana: Wiley Publishing, Inc., Copyright 2009. pp. 16

[5] National Institute of Standards and Technology. *The NIST Definition of Cloud Computing*. < http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf >. [Consulta: 21-11-2013]

[6] *Qué es la Computación Grid?*
< http://apuntescomputacion.wordpress.com/tag/grid-computing/>
[Consulta: 25 – 11- 2013]

[7] Textos científicos.com. *Arquitectura del Grid.* < http://www.textoscientificos.com/redes/computacion-grid/arquitectura > [Consulta: 12-12-2014]

[8] Textos científicos.com. *Capa de Conectividad del Grid* <http://www.textoscientificos.com/redes/computacion-grid/arquitectura/conectividad >

[9] Guía del usuario Del Grid. *Descripción General - Acceso al Grid* <http://doc.3tera.com/AppLogic31/es/User_Guide/index.htm?toc.htm?1477843.html > [Consultado: 12-01-2014]

[10] RedIRIS – FedSSH. *Documentación* <http://www.rediris.es/fedssh/index.html> [Consultado: 05-03-2014]

[11] Kantara Initiative. *Liberty Alliance.* < http://www.projectliberty.org/>. [Consultado: 15-02- 2014]

[12] Universitat Oberta de Catalunya. *Single Sing-On y Federación de identidades* <http://www.exabyteinformatica.com/uoc/Dactiloscopia/Identidad_digital/Identidad_digital_%28Modulo_4%29.pdf > [Consultado: 11-12-2013]

BIBLIOGRAFÍA

1. Eadline Douglas, *High Performance Computing For Dummies®, Sun and AMD Special Edition.* Indianapolis, Indiana: Wiley Publishing, Inc., Copyright 2009

3. Guía del usuario Del Grid. *Descripción General - Acceso al Grid.* [Online] http://doc.3tera.com/AppLogic31/es/User_Guide/index.htm?toc.htm?1477843.html

4. Kantara Initiative. *Liberty Alliance.* [Online] < http://www.projectliberty.org/>.

5. National Institute of Standards and Technology. *The NIST Definition of Cloud Computing*. [Online] http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

6. *Qué es la Computación Grid?* [Online] http://apuntescomputacion.wordpress.com/tag/grid-computing

7. RedIRIS – FedSSH. *Documentación.* [Online] http://www.rediris.es/fedssh/index.html

8. Textos científicos.com. *Arquitetura del Grid.* [Online] http://www.textoscientificos.com/redes/computacion-grid/arquitectura

9. Universitat Oberta de Catalunya. *Single Sing-On y Federación de identidades.* [Online] http://www.exabyteinformatica.com/uoc/Dactiloscopia/Identidad_digital/Identidad_digital_%28Modulo_4%29.pdf

ANEXOS

ANEXO A: CONFIGURACION SERVIDOR IDP SHIBBOLETH

1) INSTALAR APACHE Y MODULO JK

apt-get install apache2 libapache2-mod-jk

2) INSTALAR TOMCAT6

apt-get install tomcat6

#Preparar tomcat. Descargar el siguiente archivo y colocarlo en el directorio lib de tomcat
cd /usr/share/tomcat6/lib
wget
https://build.shibboleth.net/nexus/content/repositories/releases/edu/internet2/middleware/security/tomcat6/tomcat6-dta-ssl/1.0.0/tomcat6-dta-ssl-1.0.0.jar

#Instalar un servlet de Java (contenedor) para correr el IdP. Abrir el archivo de configuración de tomcat
vim /etc/tomcat6/server.xml

#Descomente la línea:
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />

#Y agregar el siguiente conector:

```
<Connector port="8443"
        protocol="org.apache.coyote.http11.Http11Protocol"

SSLImplementation="edu.internet2.middleware.security.tomcat6.DelegateToApplic
ationJSSEImplementation"
        scheme="https"
        SSLEnabled="true"
        clientAuth="want"
        keystoreFile="/opt/shibboleth-idp/credentials/idp.jks"
        keystorePass="changeit" />
```

Comente el conector asi:

```
<!--
   <Connector port="8080" protocol="HTTP/1.1"
              connectionTimeout="20000"
              URIEncoding="UTF-8"
              redirectPort="8443" />
-->
```

3) INSTALAR EL IDP DE SHIBBOLETH

```
cd /root
export JAVA_HOME="/usr/lib/jvm/java-6-openjdk-i386"
wget http://shibboleth.net/downloads/identity-provider/2.3.8/shibboleth-
identityprovider-2.3.8-bin.zip
unzip shibboleth-identityprovider-2.3.8-bin.zip



#Darle acceso a la página Status del servidor.
vim shibboleth-identityprovider-2.3.8/src/main/webapp/WEB-INF/web.xml
```

#Modificar el archivo web.xml y cerca en la line 109. Ponerle un pool de IP's
<param-value>127.0.0.1/32 ::1/128 192.168.66.251/24</param-value>

#Descomentar en el archivo las siguientes lineas

<! -- Uncomment if you want form-based auth managed by the container -->
<login-config> <auth-method>FORM</auth-method> <realm-name>IdP Password Authentication</realm-name> <form-login-config>
<form-login-page>/login.jsp</form-login-page> <form-error-page>/login-error.jsp</form-error-page> </form-login-config> </login-config>

#Instalar el IDP-Shibboleth
cd shibboleth-identityprovider-2.3.8
./install.sh

#se observa la siguiente salida
Se observa una salida como la siguiente

Buildfile: src/installer/resources/build.xml

install:
!!
Be sure you have read the installation/upgrade instructions on the Shibboleth website before proceeding.
!!
Where should the Shibboleth Identity Provider software be installed?
[/opt/shibboleth-idp]
/opt/shibboleth-idp

What is the fully qualified hostname of the Shibboleth Identity Provider server? [idp.example.org]
192.168.66.251 (SELECCIONE EL NOMBRE FQDN DEL IDP)
A keystore is about to be generated for you. Please enter a password that will be used to protect it.
xxxxxxxx (SELECCIONE LA CLAVE DEL KEYSTORE DEL IDP)
Updating property file: /usr/local/src/shibboleth-identityprovider-2.3.8/src/installer/resources/install.properties
Created dir: /usr/local/idp
Created dir: /usr/local/idp/bin
Created dir: /usr/local/idp/conf
Created dir: /usr/local/idp/credentials
Created dir: /usr/local/idp/lib
Created dir: /usr/local/idp/lib/endorsed
Created dir: /usr/local/idp/logs
Created dir: /usr/local/idp/metadata
Created dir: /usr/local/idp/war
Generating signing and encryption key, certificate, and keystore.
Copying 5 files to /usr/local/idp/bin
Copying 8 files to /usr/local/idp/conf
Copying 1 file to /usr/local/idp/metadata
Copying 51 files to /usr/local/idp/lib
Copying 5 files to /usr/local/idp/lib/endorsed
Copying 1 file to /usr/local/src/shibboleth-identityprovider-2.3.8/src/installer
Building war: /usr/local/src/shibboleth-identityprovider-2.3.8/src/installer/idp.war
Copying 1 file to /usr/local/idp/war
Deleting: /usr/local/src/shibboleth-identityprovider-2.3.8/src/installer/web.xml
Deleting: /usr/local/src/shibboleth-identityprovider-2.3.8/src/installer/idp.war

BUILD SUCCESSFUL

Total time: 2 minutes 43 seconds

4) DARLE PERMISO A TOMCAT A LAS CARPETAS RAIZ, LOGS Y METADATA. COPIAR EL DIRECTORIO ENDORSED DE IDP EN TOMCAT

```
chown tomcat6:tomcat6 /opt/shibboleth-idp/
chown tomcat6:tomcat6 /opt/shibboleth-idp/logs/
chown tomcat6:tomcat6 /opt/shibboleth-idp/metadata/
cp -a /opt/shibboleth-idp/lib/endorsed/ /usr/share/tomcat6/
```

5) DESPLEGAR EL IDP USANDO UN FRAGMENTO DE DESPLIEGUE DE CONTEXTO. CREAR EL ARCHIVO

```
vim /etc/tomcat6/Catalina/localhost/idp.xml

#Agregar lo siguiente al archivo
<Context docBase="/opt/shibboleth-idp/war/idp.war"
        privileged="true"
        antiResourceLocking="false"
        antiJARLocking="false"
        unpackWAR="false"
        swallowOutput="true" />
```

6) CONFIGURAR APACHE2

```
vim /etc/apache2/sites-available/idp-SSO

#Agregar lo siguiente al archivo

<VirtualHost 192.168.85.131:443>
```

```
  ServerName 192.168.85.131
<Proxy ajp: //localhost:8009>
       Order deny, allow
       Allow from all
   </Proxy>

  ServerSignature Off
  SSLEngine on
  SSLProxyEngine On
  SSLCertificateKeyFile /etc/ssl/private/chave-apache.key
  SSLCertificateFile /etc/ssl/certs/certificado-apache.crt
  DocumentRoot /var/www/vazio/

  <Directory /var/www/vazio/>
    Options -Indexes -FollowSymLinks -MultiViews
    AllowOverride None
    Order deny, allow
    Deny from all
  </Directory>

  ProxyRequests Off
  ProxyPreserveHost On
  ProxyPass / ajp: //localhost:8009/
  ProxyPassReverse / ajp://localhost:8009/
  CustomLog /var/log/apache2/access-idp-443.log combined
  LogLevel warn
  ErrorLog /var/log/apache2/error-idp-443.log
</VirtualHost>
```

7) CREAR EL ARCHIVO IDP.CONF EN /ETC/APACHE2/CONF.D/

vim /etc/apache2/conf.d/idp.conf

#Agregar las siguientes lineas
#JkWorkersFile /etc/libapache2-mod-jk/workers.properties
JkShmFile /var/run/apache2/jk-runtime-status.Shm
JkLogFile /var/log/apache2/mod_jk.log
JkLogLevel info
#JkMount /idp/* ajp13_worker

#Generar los certificados de apache cliente

openssl genrsa 2048 -config openssl.cnf > /etc/ssl/private/chave-apache.key

openssl req -new -x509 -nodes -days 1095 -sha1 -key /etc/ssl/private/chave-apache.key -set_serial 00 -config openssl.cnf > /etc/ssl/certs/certificado-apache.crt

chown root /etc/ssl/private/chave-apache.key /etc/ssl/certs/certificado-apache.crt

#Crear el directorio y desplegar el sitio web
mkdir /var/www/vazio/
a2dissite default
a2ensite idp-SSO
a2enmod ssl
a2enmod proxy
a2enmod proxy_ajp

a2enmod jk

8) CONFIGURAR IDP-SHIBBOLETH

#comente la sección que se refiere a RemoteUser y habilite la sección UsernamePassword. El archivo final debe tener la siguiente configuración:

vim /opt/shibboleth-idp/conf/handler.xml

```
<!-- Login Handlers -->
<!--
   <ph:LoginHandler xsi:type="ph:RemoteUser">

<ph:AuthenticationMethod>urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified</
ph:AuthenticationMethod>
   </ph:LoginHandler>
-->

<!--  Username/password login handler -->
ph:LoginHandler xsi:type="ph:UsernamePassword"
                jaasConfigurationLocation="file:///opt/shibboleth-idp/conf/login.config">

<ph:AuthenticationMethod>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProt
ectedTransport</ph:AuthenticationMethod>
   </ph:LoginHandler>
```

#Configurar la conexion al LDAP modificando el siguiente archivo
vim /opt/shibboleth-idp/conf/login.config

#con la siguientes lineas

```
edu.vt.middleware.ldap.jaas.LdapLoginModule required
    host="192.168.66.251:389"
    base="ou=people,dc=uis,dc=edu,dc=co"
    serviceUser="cn=admin,dc=uis,dc=edu,dc=co"
    serviceCredential="1234"
    subtreeSearch="true"
    ssl="false"
    tls="false"
    userField="uid";
```

```
#agregar certificado de confianza de LDAP
keytool -import -trustcacerts -alias "Local-CA" -file /opt/shibboleth-
idp/credentials/_local-ca_.crt -keystore $JAVA_HOME/jre/lib/security/cacerts
```

9) DEFINIR LOS DATOS DEL SERVIDOR LDAP Y LOS ATRIBUTOS A COMPARTIR. EDITAR EL ARCHIVO

```
vim /opt/shibboleth-idp/conf/attribute-resolver.xml
```

```
#Descomentar las siguientes líneas. (Definición del email como atributo)
<resolver:AttributeDefinition xsi:type="ad:Simple" id="email"
sourceAttributeID="mail">
<resolver:Dependency ref="myLDAP" />
    <resolver:AttributeEncoder xsi:type="enc:SAML1String"
name="urn:mace:dir:attribute-def:mail" />
    <resolver:AttributeEncoder xsi:type="enc:SAML2String"
name="urn:oid:0.9.2342.19200300.100.1.3" friendlyName="mail" />
</resolver:AttributeDefinition>
```

#Y descomentar las siguientes líneas al final (datos del servidor LDAP)

```
<resolver:DataConnector id="myLDAP" xsi:type="dc:LDAPDirectory"
ldapURL="ldaps://192.168.1.6:389"
baseDN="ou=people,dc=uis,dc=edu,dc=co"
principal="cn=admin,dc=uis,dc=edu,dc=co"
principalCredential="xxxxxx">
<dc:FilterTemplate>
<![CDATA[
(uid=$requestContext.principalName)
]]>
</dc:FilterTemplate>
</resolver:DataConnector>
```

#Definir los atributos que se quieren publicar. Editar el archivo

```
vim /opt/shibboleth-idp/conf/attribute-filter.xml
```

```
<afp:AttributeFilterPolicy id="attributeToAnyone">
<afp:PolicyRequirementRule xsi:type="basic:ANY"/>
        <afp:AttributeRule attributeID="email">
        <afp:PermitValueRule xsi:type="basic:ANY"/>
       </afp:AttributeRule>
</afp:AttributeFilterPolicy>
```

#Arreglar jk.conf para que apache no de error

```
cd /etc/apache2/mods-enabled/
mv jk.conf prueba
```

#Hacer un link simbolico en shibboleth_home

```
cp /media/D8F7-D087/servlet-api.jar $CATALINA_HOME/lib/
ln -s $CATALINA_HOME/lib/servlet-api.jar /opt/shibboleth-idp/lib/
```

10) REINICIAR SERVICIOS DE APACHE Y TOMCAT

service apache2 restart
service tomcat6 restart

Posibles errores:

Apr 2, 2014 8:00:48 PM org.apache.catalina.core.StandardContext start
SEVERE: Error listenerStart
Apr 2, 2014 8:00:48 PM org.apache.catalina.core.StandardContext start
SEVERE: Context [/idp] startup failed due to previous errors

Solucion: Necesita conectarse al LDAP. El LDAP no es alcanzable

---

Exception in thread "main"
org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'shibboleth.HandlerManager': Initialization of bean failed; nested exception is java.lang.NoClassDefFoundError: javax/servlet/ServletRequest

---

No hay enlace simbolico en ln -s $CATALINA_HOME/lib/servlet-api.jar /opt/shibboleth-idp/lib/

ANEXO B: CONFIGURACION SERVIDOR SP SIMPLESAML

1) INSTALACION DE APACHE2 Y LIBRERIAS DE PHP

apt-get install apache2 php5 libapache2-mod-php5

2) DESCARGAR SIMPLESAMLPHP Y SE DESCOMPRIME

cd /var/
wget https://simplesamlphp.org/res/downloads/simplesamlphp-1.12.0.tar.gz

#se descomprime
tar xzf simplesamlphp-1.12.0.tar.gz

#Se cambia el nombre de la carpeta descomprimida
mv simplesamlphp-1.12.0 simplesamlphp

#Se hace una copia a las siguientes carpetas
cd simplesaml
cp -r config-templates/*.php config/
cp -r metadata-templates/*.php metadata/

3) SE CREA EL SIGUIENTE ARCHIVO EN EL SERVIDOR APACHE

vim /etc/apache2/sites-available/simplesamlphp

#Se agrega lo siguiente al archivo

```
<VirtualHost 192.168.66.252:443>
   ServerName 192.168.66.252
   SSLEngine on
   SSLCertificateKeyFile /etc/ssl/private/chave-apache.key
   SSLCertificateFile /etc/ssl/certs/certificado-apache.crt

   DocumentRoot /var/www/

   Alias /simplesaml /var/simplesamlphp/www

   CustomLog /var/log/apache2/access-sp-443.log combined
   LogLevel warn
   ErrorLog /var/log/apache2/error-sp-443.log
</VirtualHost>
```

#Se generan los certificados chave-apache.key, certificado-apache.crt

```
openssl genrsa 2048 -config openssl.cnf > /etc/ssl/private/chave-apache.key
```

```
openssl req -new -x509 -nodes -days 1095 -sha1 -key /etc/ssl/private/chave-apache.key -set_serial 00 -config openssl.cnf > /etc/ssl/certs/certificado-apache.crt
```

```
chown root /etc/ssl/private/chave-apache.key /etc/ssl/certs/certificado-apache.crt
```

4) CONFIGURAR SIMPLESAMLPHP

```
vim /var/simplesamlphp/config/config.php
```

#cambiar el dato por la contraseña que usted quiera para entrar al administrador de la aplicacion
'auth.adminpassword' => '1234',

5)Se configura los metadatos del IDP agregando las siguientes lineas en el archivo vim /var/simplesamlphp/metadata/saml20-idp-remote.php

```
$metadata['https://192.168.66.251/idp/shibboleth'] = array (
  'name' => '192.168.66.251',
  'description' => 'Shibboleth 2.3.8',
  'SingleSignOnService' =>
'https://192.168.66.251/idp/profile/SAML2/Redirect/SSO',
  'certFingerprint' => '4307846FC5F48840D11C50363FEE7F1D20BCF2E4',
  'authproc' => array(
    50 => array(
        'class' => 'core:AttributeMap',
        'oid2name',
    ),
  ),
);
```

#certFingerprint =>Este valor se obtiene en el IDP de la siguiente forma
cd /opt/shibboleth-idp/credentials/
cat idp.crt | openssl x509 -fingerprint | grep SHA1 | sed "s/^[^=]*=//g" | sed "s/://g"

#El resultado se copia en la variable &certFingerprint

ANEXO C: CONFIGURACION DE LA APLICACIÓN FEDSSH[8]

1) CREAR EL SP-HOST EN SIMPLESAMLPHP EN EL SIGUIENTE ARCHIVO

vim /var/simplesamlphp/metadata/saml20-sp-hosted.php

```
<?php
$metadata ['fedssh'] = array (
  'host' => '192.168.66.252',

);
```

---------- En la bibliografía colocar que de acá para abajo se basó en la documentación de la aplicacion fedssh disponible en la página web http://rediris.es/fedssh----------

2) DESCARGAR Y DESCOMPRIMIR LA APLICACION FEDSSH

```
cd /var/
wget http://forja.rediris.es/frs/download.php/1278/fedssh-web-1.0.zip
unzip fedssh-web-1.0.zip
mv fedssh-web-1.0 fedssh
```

3) GENERAR EL ACCESO A APACHE DE LA APLICACION WEB

[8] La información de este anexo se basó de la documentación de RedIRIS. http://www.rediris.es/fedssh/doc.html

#en el mismo virtualhost de simplesaml se escriben estas lineas
vim /etc/apache2/sites-available/simplesamlphp

Alias /fedssh /var/fedssh/web/

4) SE CONFIGURA LA APLICACION WEB PARA CONECTAR AL LDAP

cd /fedssh
vim siledap_api_lib/LdapConf.php

$LDAP_SERVER_NAME="192.168.66.251";
$LDAP_SERVER_PORT=389;
$LDAP_USER="cn=admin,dc=uis,dc=edu,dc=co";
$LDAP_PASS="xxxx";
$LDAP_DN_BASE="dc=ssh,dc=uis,dc=edu,dc=co";

$LOWERCASE_ATTRIBUTES = false;

$LDAP_SERVER_NAME: es la dirección del servidor LDAP.
$LDAP_SERVER_PORT: es el puerto del servidor LDAP.
$LDAP_USER: es el DN de usuario para conectarse a dicho servidor.
$LDAP_PASS: es la contraseña de dicho usuario.
$LDAP_DN_BASE: es el DN base donde se añadirán todas las entradas que la aplicación web necesita. Esta entrada debe existir antes de utilizar la aplicación.
$LOWERCASE_ATTRIBUTES: indica si el servidor LDAP es sensible a minúsculas/mayúsculas. Tocar sólo si es necesario.

Configuración necesaria para la aplicación web.
vim config.php

```
# Directorio de FedSSH para el servidor HTTP:

$sshfed_folder = "/var/fedssh/web/";

#Directorio donde se encuentra la librería siLeDAP:

$siledap_folder = "/var/fedssh/siledap_api_lib/";

#Módulo de autenticación:

$auth_module = "simplesamlphp";
$sSp_module_config = array(
        'path' => '/var/simplesamlphp/www/',
        'mode' => 'saml2'
);
```

5) CONFIGURACIÓN DE LOS SERVIDORES QUE SE CONECTARAN A LA APLICACIÓN

vim /policies/example.xml

```
<SSHServer host="192.168.66.252" port="22">
        <CheckStatus value="urn:mace:rediris.es:check:cron:2/0" />
        <Policies>
        <Policy user="root" timeout="20">
                <Attributes check="none" >
                        <Attribute name="cn" value="John" />
                </Attributes>
        </Policy>
```

</Policies>
</SSHServer>

Donde,

El atributo host de <SSHServer> representa la dirección o nombre de la máquina.
El atributo port de <SSHServer> representa el puerto SSH de la máquina.
El elemento CheckStatus define cómo está configurado el servicio SSH en esa máquina para conectarse con FedSSH. Los posibles valores

en su atributo value son:
urn:mace:rediris.es:check:none: esta URN indica que el SSH de la máquina está parcheado y que se conecta directamente al directorio cada vez que alguien intenta acceder por SSH para obtener sus claves públicas.

El prefijo urn:mace:rediris.es: de dichas URNs puede ser cambiada a cualquier otra, cambio que debe ser reflejado también en la configuración de los componentes correspondientes.

Dentro de <Policies> añadimos un elemento <Policy> por cada uno de los usuarios a los que se podrá acceder por SSH, donde:
El atributo user de <Policy> especifica la política de autorización para acceder como dicho usuario a la máquina.
El atributo timeout de <Policy> especifica durante cuánto tiempo la clave pública del usuario será válida en el directorio en el momento que dicho usuario lo solicite.
El elemento <Attributes> dentro un elemento <Policy> define la lógica, a través de su atributo check, a la hora de comprobar los atributos del usuario:

all: se permitirá el acceso en el caso de que cumplan todos los atributos que se especifican dentro del mismo.

any: se permitirá el acceso en el caso de que cumpla alguno de los atributos

que se especifican dentro del mismo.

none: se permitirá el acceso en el caso de que no cumpla ninguno de los atributos que se especifican dentro del mismo.

El elemento <Attribute> dentro un elemento <Attributes> define una regla sobre un atributo del usuario, donde name indica el nombre del atributo y value el valor que debe tener.

Posibles errores:

En el archivo /var/fedssh/web/api/AuthModules/sSpAuth.php

cambiar la línea 6 por

require_once("/var/simplesamlphp/www/_include.php");

ANEXO D: PARCHE Y CONFIGURACION OPENSSH

1) DESCARGA DEL PATCH Y CODIGO FUENTE DEL SSH

```
wget http://distfiles.gentoo.org/distfiles/openssh-lpk-5.9p1-0.3.14.patch.gz
gzip -d openssh-lpk-5.9p1-0.3.14.patch.gz
apt-get build-dep openssh
apt-get install libldap2-dev quilt
apt-get source openssh
cd openssh-5.9p1
patch < ../openssh-lpk-5.9p1-0.3.14.patch
dpkg-source --commit
```

```
# Eliminar la linea 234 porque tiene un bug
vim ./auth-rsa.c
quilt refresh
```

```
# Editar el siguiente archivo
vim debian/rules
 --- a/rules 2012-04-02 10:38:04.000000000 +0000
 +++ b/rules 2012-06-12 21:46:43.000000000 +0000
 @@ -81,6 +81,7 @@

  # The deb build wants xauth; the udeb build doesn't.
  confflags += --with-xauth=/usr/bin/xauth
 +confflags += --with-ldap
```

```
 confflags_udeb += --without-xauth

 # Default paths. The udeb build has /usr/bin/X11 and /usr/games removed.
@@ -93,6 +94,7 @@
 cflags := $(default_cflags)
 cflags += -DLOGIN_PROGRAM=\"/bin/login\" -DLOGIN_NO_ENDOPT
 cflags += -DSSH_EXTRAVERSION=\"$(SSH_EXTRAVERSION)\"
+cflags += -DWITH_LDAP_PUBKEY
 cflags_udeb := -Os
 cflags_udeb += -DSSH_EXTRAVERSION=\"$(SSH_EXTRAVERSION)\"
 confflags += --with-cflags='$(cflags)'
```

```
#anexar libldap2-dev a Build-Depends de debian/control.
vim debian/control
(snip)
Build-Depends: ..., libldap2-dev
(snip)
```

2) GENERAR PAQUETE E INSTALAR SSH

dpkg -i openssh-client_5.9p1-5ubuntu1+cust1_amd64.deb openssh-server_5.9p1-5ubuntu1+cust1_amd64.deb

3) CONFIGURAR OPENSSH

vim /etc/ssh/sshd_config

UseLPK yes

LpkServers ldap://192.168.66.251/

LpkUserDN cn=admin,dc=uis,dc=edu,dc=co

LpkGroupDN ou=People,dc=uis,dc=edu,dc=co

LpkForceTLS no